DOCUMENT MANAGEMENT SYSTEM

BY

OYESUNLE OLAMILEKAN
MATRIC NO: 20203181

IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE (B.Sc) in COMPUTER SCIENCE

THE DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF PHYSICAL SCIENCES
FEDERAL UNIVERSITY OF AGRICULTURE, ABEOKUTA.

FEBUARY, 2025

CHAPTER ONE

BACKGROUND OF STUDY

Today, technology has significantly improved efficiency in various aspects of life. It has transformed communication, enhanced mobility, and streamlined numerous processes in this fast-evolving digital world. The advancement of web-based systems has made people more reliant on technology, preferring quick and easy access to essential information at their convenience.

According to Miller (2019), traditional document management systems often face challenges like document loss, poor version control, security vulnerabilities, and inefficiencies in data retrieval. These issues not only hinder operational efficiency but also elevate the risk of errors and security breaches. As a result, there has been a growing need for more effective solutions, leading to the development of digital Document Management Systems (DMS) aimed at improving security, enhancing document retrieval, and streamlining workflows.

In today's digital era, efficient document management is crucial for organizations and individuals to store, retrieve, and process information seamlessly. Traditional paper-based document handling poses quite a few challenges. To address these limitations, Document Management Systems (DMS) have emerged as essential tools for digitizing, organizing, and managing documents securely.

The Document Management System is a comprehensive software solution designed to streamline document handling and workforce management within an organization. It simplifies document submission, secure storage, retrieval, version control, and access management while enforcing strong validation, encryption, and data protection protocols.

The system allows users to save their personal information, employment history, and important documents, which they can easily access and reuse when applying within or outside the organization. This ensures a seamless application process which reduces the need for repetitive data entry.

The system integrates a dedicated Employee Management module, allowing organization to track and manage team members' profiles, contact details, employment records, and relevant documents. Employees can apply for internal and external job roles directly through the system, with their professional information, document attachments, and status updates securely maintained.

`

The use of cloud-based storage and structured workflows integrated to the system ensures flexibility, accessibility, and operational efficiency, making it easier for organizations to manage both corporate documents and human resources from a unified, secure platform.

Additionally, the integration of AI enhances the platform's capabilities by providing intelligent suggestions when filling out job application forms. Based on users' saved profiles, employment records, and uploaded documents, the integrated AI can recommend jobs, generate personalized cover letters based on the user's profile, and provide suggestions during applications which improves both the speed and quality of applications while offering a tailored user experience.

1.2 PROBLEM STATEMENT

In today's digital landscape, job applicants and organizations face persistent challenges in managing documents and employment records efficiently. Existing systems often lack secure storage, version control, and integrated workflows, making document organization, retrieval, and application tracking cumbersome. Applicants frequently struggle with outdated platforms that do not support seamless job application processes, real-time status updates, or AI-driven suggestions to improve application quality. Additionally, the absence of centralized employee management leaves companies with scattered records and limited visibility into staff profiles and document histories. These inefficiencies result in delays, missed opportunities, and poor user experiences which underscores the need for a modern, secure, and intelligent Document Management System that simplifies document handling, streamlines job applications, and empowers users with AI-powered form assistance and application tracking.

1.3 AIMS & OBJECTIVES

Aims:

The aim of this project is to develop an integrated document management system that streamlines document handling and workforce management through secure, user friendly and AI-enhanced functionalities.

`

Objectives:

- To develop AI-driven form assistance capabilities that enhance user experience during job applications.
- To implement a secure, cloud-based storage system for reliable document management.
- To integrate a feedback mechanism that provides users with real-time updates on application status.
- To design and incorporate an Employee Management Module for streamlined workforce data management.
- To establish robust authentication and role-based authorization controls for safeguarding sensitive information.
- To develop a responsive, cross-device compatible system ensuring accessibility across multiple platforms.

1.4 SCOPE OF STUDY

This study aims to develop a Document Management System (DMS) to enhance document handling, job application processes, and workforce management. The system will enable secure document management with version control and structured workflows. It will include an Employee Management Module for maintaining detailed employee records and facilitating job applications using stored information. AI integration will provide intelligent suggestions for application forms, while real-time updates, document validation, secure authentication, and cloud storage will ensure flexibility and accessibility. The system will be optimized for responsive use across various devices, ensuring a seamless user experience.

1.5 SIGNIFICANCE OF STUDY

This study is significant as it addresses critical limitations in existing document management systems, particularly in the context of job applications and workforce administration. By developing a secure, user-friendly, and intelligent Document Management System (DMS), the project aims to improve how job applicants and organizations manage, submit, and employment-related documents and records. The proposed system introduces enhanced features such as real-time application status updates, AI-powered form assistance, and cloud-based storage, which collectively streamline the job application process and promote operational efficiency. The integration of artificial intelligence ensures applicants receive intelligent suggestions when completing application forms, reducing input errors, saving time,

and increasing application accuracy based on previously saved information and qualifications.

Additionally, the Employee Management Module allows organizations to efficiently manage employee profiles, employment histories, contact information, and associated documents. This contributes to improved record-keeping practices, secure data handling, and structured workflows within companies.

By providing a unified platform for document management, job application tracking, and employee record administration, the system enhances the overall experience for job applicants, employees, and organizational administrators. Furthermore, the implementation of robust security measures, document version control, and validation protocols ensures data integrity and protection against unauthorized access.

Ultimately, this study contributes to the advancement of modern recruitment and workforce management processes by offering a comprehensive, intelligent, and secure solution that aligns with contemporary digital workplace needs.

1.6 MOTIVATION OF THE STUDY

The motivation for this study arises from the challenges faced by job applicants and organizations in managing employment-related documents and job applications using traditional, disconnected systems.
These challenges include repetitive data entry, insecure document submissions, lack of application status visibility, disorganized employee records, and administrative inefficiencies.
This project aims to address these issues by providing a unified, modern, and secure platform that simplifies document management and workforce administration. The integration of AI will streamline the application process, offering data-driven assistance to improve speed and accuracy. The inclusion of an Employee Management Module will support secure, accurate, and accessible employee records, contributing to enhanced operational efficiency and user experience in digital workplaces.

CHAPTER TWO

LITERATURE REVIEW

2.0 OVERVIEW OF DOCUMENT MANAGEMENT SYSTEMS (DMS)

Document Management Systems (DMS) are integrated platforms designed to streamline the management of organizational documents and workforce records. These systems not only ensure efficient storage, retrieval, and version control of documents but also provide tools for managing employee information, employment histories, and related documents securely. Modern DMS enhance productivity by incorporating cloud-based storage, secure access control, and AI-driven suggestions for form completion, particularly during job applications.

Such systems support structured workflows, allowing users to save personal data and reuse it across multiple applications, while offering real-time updates on application statuses. As digital transformation advances, DEMS increasingly include intelligent document handling, automated validation checks, and seamless application processes to improve operational efficiency and user satisfaction (Kumar & Li, 2022). Implementing a robust document management system is crucial for organizations to ensure efficient storage, retrieval, and sharing of documents, while also maintaining data integrity and security" (Sharma & Patel, 2020).

2.1 CHALLENGES IN TRADITIONAL DOCUMENT MANAGEMENT

Traditional document management systems often rely on paper-based processes, which are prone to inefficiencies, such as slow retrieval times, data loss, and difficulty in sharing information across departments. Additionally, physical storage requires significant space and incurs high costs for maintenance and document retrieval. The limitations of traditional methods highlight the need for automated, digital solutions to streamline document management and reduce operational costs.

2.2 THE ROLE OF CLOUD STORAGE IN DOCUMENT MANAGEMENT SYSTEMS (DMS)

Cloud-based storage solutions have become an essential component of modern Document Management Systems. By leveraging the cloud, DMS can offer scalable storage options, making it easier to manage growing volumes of data. Cloud storage provides increased accessibility, as users can access documents remotely from any device, improving collaboration and flexibility. Furthermore, cloud solutions often come with built-in security

features like encryption, backup, and disaster recovery options, addressing concerns over data security.

2.3 USE OF DJANGO AND DATABASE TECHNOLOGIES IN DOCUMENT MANAGEMENT SYSTEMS

Modern Document Management Systems (DMS) leverage technologies like Django and robust databases to improve functionality and efficiency. Django, a Python-based web framework, enables rapid development of secure and scalable web applications, allowing seamless document access, search features, and customizable workflows (Smith, 2020). Databases such as PostgreSQL, MySQL, and MongoDB manage large volumes of structured and unstructured data, ensuring efficient document storage, retrieval, and version control (Evans & Green, 2019). Additionally, cloud platforms like AWS provide scalable storage solutions, enhancing accessibility and collaboration. The integration of these technologies ensures improved data security, scalability, and user experience in DMS (Jackson & Lee, 2018).

2.4 SECURITY FEATURES IN DOCUMENT MANAGEMENT SYSTEMS

The security of documents in a DMS is of paramount importance, especially when dealing with sensitive or confidential information. Features such as role-based access control, encryption, and audit trails are critical in protecting data and ensuring only authorized personnel can access certain documents. Role-based access control allows administrators to define user permissions based on roles, ensuring that only authorized individuals can perform specific actions, such as editing or deleting documents. Encryption ensures that data remains secure both during transmission and while stored in the system (Evans & Green, 2019). Security in document management systems must address access control, encryption, and audit trails to protect sensitive organizational information. (Mohammad & Lee, 2018)

2.5 THE ROLE OF AI IN DOCUMENT MANAGEMENT SYSTEMS (DMS)

Artificial Intelligence (AI) has become a vital component in modern Document Management Systems (DMS), transforming how documents are handled, organized, and accessed. AI enhances document classification, metadata extraction, and automated tagging, reducing manual effort and minimizing human error. In job application systems, AI can suggest personalized content for fields like cover letters and skills, drawn from users' stored profiles and past interactions, improving

accuracy and efficiency (Soni, P., & Jain, M. (2017).). Intelligent form field assistance using machine learning can significantly enhance user experience by providing real-time suggestions and error detection, thereby improving the efficiency and accuracy of form filling processes, (Meyer, R., & Gupta, K. (2020).)

Additionally, AI supports predictive search, intelligent form filling, and real-time status tracking, enabling applicants to receive timely feedback and tailored job recommendations based on their qualifications. By integrating AI-driven features, DMS not only increase productivity but also enhance user experience and decision-making processes (Rahman & Singh, 2021).

2.6 THE ROLE OF EMPLOYEE MANAGEMENT IN DOCUMENT MANAGEMENT SYSTEMS (DMS)

Employee management plays an essential role in modern Document Management Systems (DMS) by centralizing the storage and control of employee records, credentials, and work-related documents within a secure and organized environment. Integrating employee management functionalities into a DMS ensures that organizations can efficiently track staff profiles, employment histories, and related documentation while maintaining compliance with data security standards.

Such systems facilitate quick retrieval of employee documents for operational and administrative tasks, including internal job applications and performance reviews. Moreover, combining employee management with document handling enhances workforce transparency, data accuracy, and operational efficiency (Adeyemi & Thomas, 2020).

2.7 USER EXPERIENCE AND INTERFACE DESIGN IN DMS

An intuitive and user-friendly interface is crucial for the success of a Document Management System. Users will be able to easily navigate the system, track documents, and perform actions such as uploading, editing. A well-designed DMS provides feedback mechanisms to inform users about system actions (e.g., document upload completion or error messages). This ensures a smooth interaction and minimizes frustration. Furthermore, features such as document preview and integration with existing productivity tools enhance the overall user experience (Smith, 2020).

CHAPTER THREE

METHODOLOGY

3.0 INTODUCTION

This chapter provides detailed examination of the methodology employed in the design and implementation of the proposed Document Management System.The development process adhered to a systematic and iterative framework, ensuring that both functional and non-functional requirements are thoroughly addressed.

This chapter commences with an exploration of the design considerations, emphasizing the factors that is set to shape the system's architecture, user interface, data security protocols, workflow management,and feature integration, such as AI-driven form assistance and cloud-based storage. These considerations were pivotal in ensuring that the system will remain secure, scalable,and user-friendly.

Subsequently, the proposed system architecture is outlined and it illustrates how various components and module including document management, employee records, job application tracking, and AI assistance are interconnected.The architecture explains the logical and physical arrangement of the system's components, highlighting the data flow and interaction between users and the platform.

Finally, this chapter describes the system algorithms and formal methods utilized in the implementation of core processes within the system. These algorithms govern essential operations such as authentication and authorization, secure document upload, real-time application status updates, intelligent form suggestions, and employee management. The formal methods ensure the accuracy, reliability, and security of these operations, thereby enhancing overall system efficiency.

This structured methodology laid the groundwork for the entire development process, ensuring that the proposed solution achieves its intended objectives while upholding high standards of performance, usability, scalability and data security..

3.1 DESIGN CONSIDERATION

In developing the Document Management System, several essential factors are prioritized to ensure the platform is secure, efficient, and scalable. These include:

1. AI-Driven Form Assistance:

The Document Manage System aims to integrate AI-powered suggestions to assist users while filling out job application forms. It automatically suggest skills, cover letter content, and suitable job positions based on users' saved information and past applications, reducing input errors and enhancing user experience. This design consideration will ensure that users can complete applications more efficiently and accurately, and minimizing repetitive data entry. This intelligent assistance also contributes to higher user satisfaction and engagement, making the application process smoother and more intuitive.

2. Cloud-Based Storage Integration:

The platform will be structured with designs to guarantee data accessibility and secure storage, the system utilizes a cloud-based storage infrastructure. This will allow users to upload, retrieve, and manage their documents and records from any location while ensuring data security, backup, and disaster recovery. This flexibility will support remote access and improves productivity for both employees and employers. In addition to accessibility, the cloud architecture aims to enhance security through encrypted data transmission, and access control mechanisms, ensuring that only authorized users can interact with sensitive documents.

3. Performance and Efficiency:

The system plans to undergo designs for fast response times and efficient resource management. Lightweight technologies and optimized database queries will ensure quick document retrieval, seamless form submissions, and reliable multi-user operations without system slowdowns. To further enhance performance, the architecture will include caching mechanisms for frequently accessed data and asynchronous task handling to manage background processes without blocking user actions. Efficient indexing strategies will be adopted to reduce query execution time, while front-end optimization technique such as minimizing script sizes and lazy-loading assets will contribute to a smoother user experience.

4. Security and System Scalability:

The platform will be built with a modular, scalable architecture and robust security protocols. It will ensure data encryption, secure data handling through CSRF protection, server-side validation enforcement, and input sanitization while maintaining capacity for future growth in user volume, data storage, and new feature integration without degrading performance. Role-based access control (RBAC) will be implemented to ensure only authorized users can access or modify sensitive data. Additionally, activity logging and audit trails will be incorporated for monitoring and accountability. These measures will collectively deliver a future-proof system that is resilient, adaptable, and secure.

`

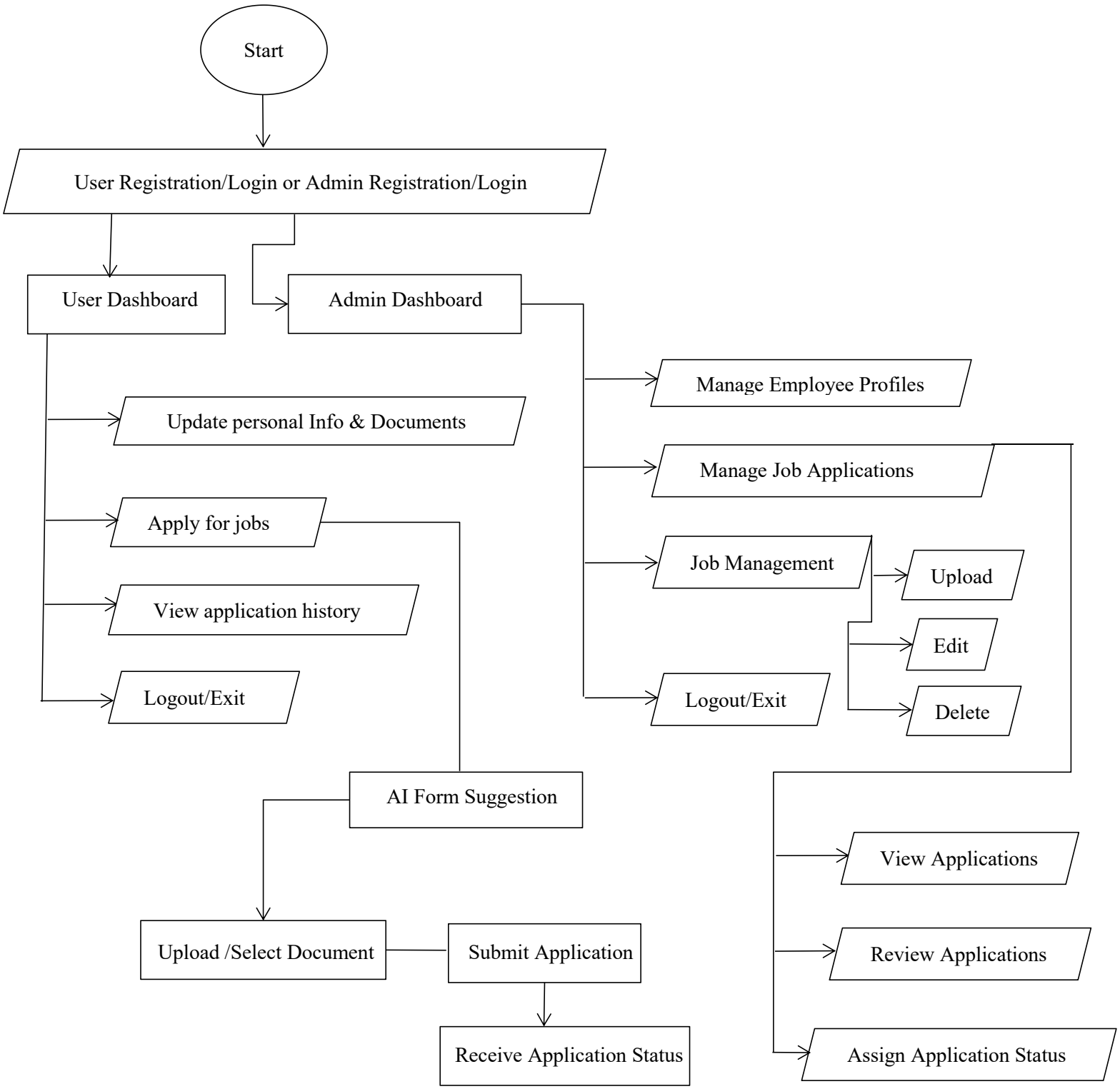


Fig. 1 Flow Chart for the Proposed Document Management System

`

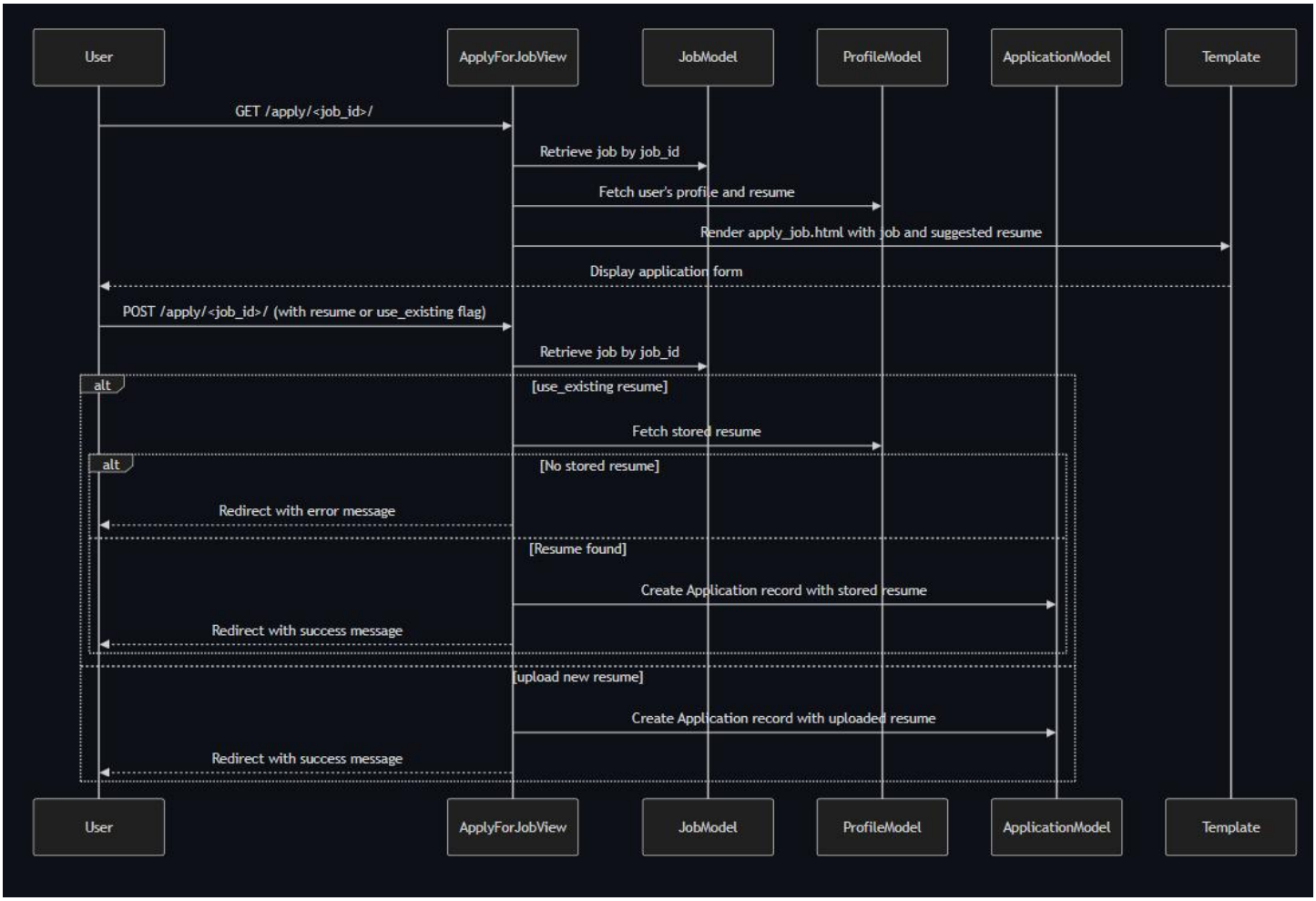


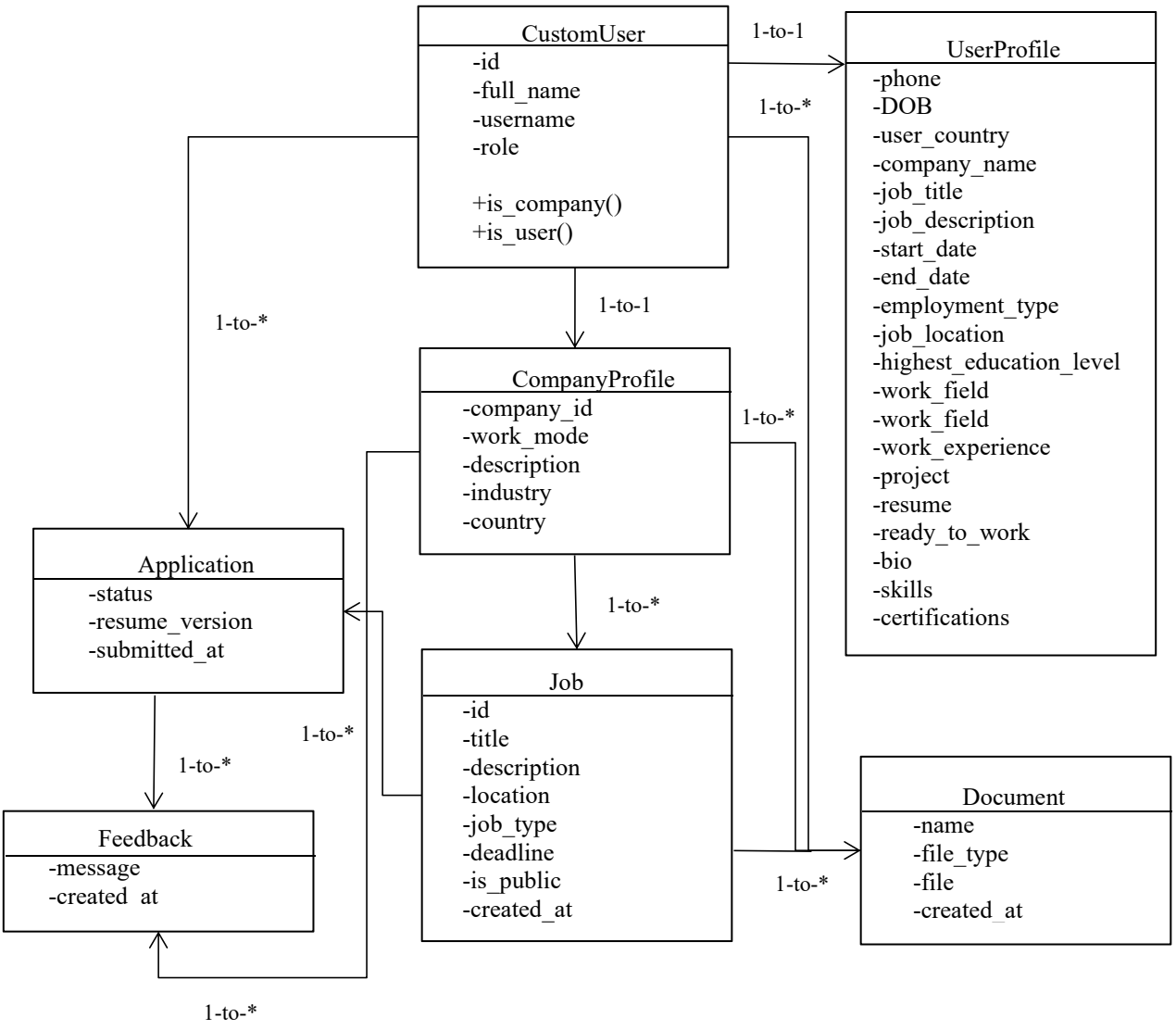
Fig. 2 Sequence diagram for Application in the Proposed Document Management System

The sequence diagram outlines the job application process as defined in the formalism (3.4). A logged-in user (from U) initiates an application, triggering auth(u_i, p) for credential validation. The system uses suggest(f_i) to offer AI-powered resume suggestions based on saved profile data (D).

If the user opts for an existing resume, apply_job(u_i, j_j, existing_resume) is executed; otherwise, a new file is uploaded using upload(u_i, d_j). If no resume is found, the system flags an error linked to AI Error Detection & Correction. A valid submission creates a new record in A = {a_1, a_2, ..., a_k}, associating the user, job, resume, and status ('pending'). The process concludes with a success message and redirection, completing the logical flow under 3.5 Job Application Process.

`

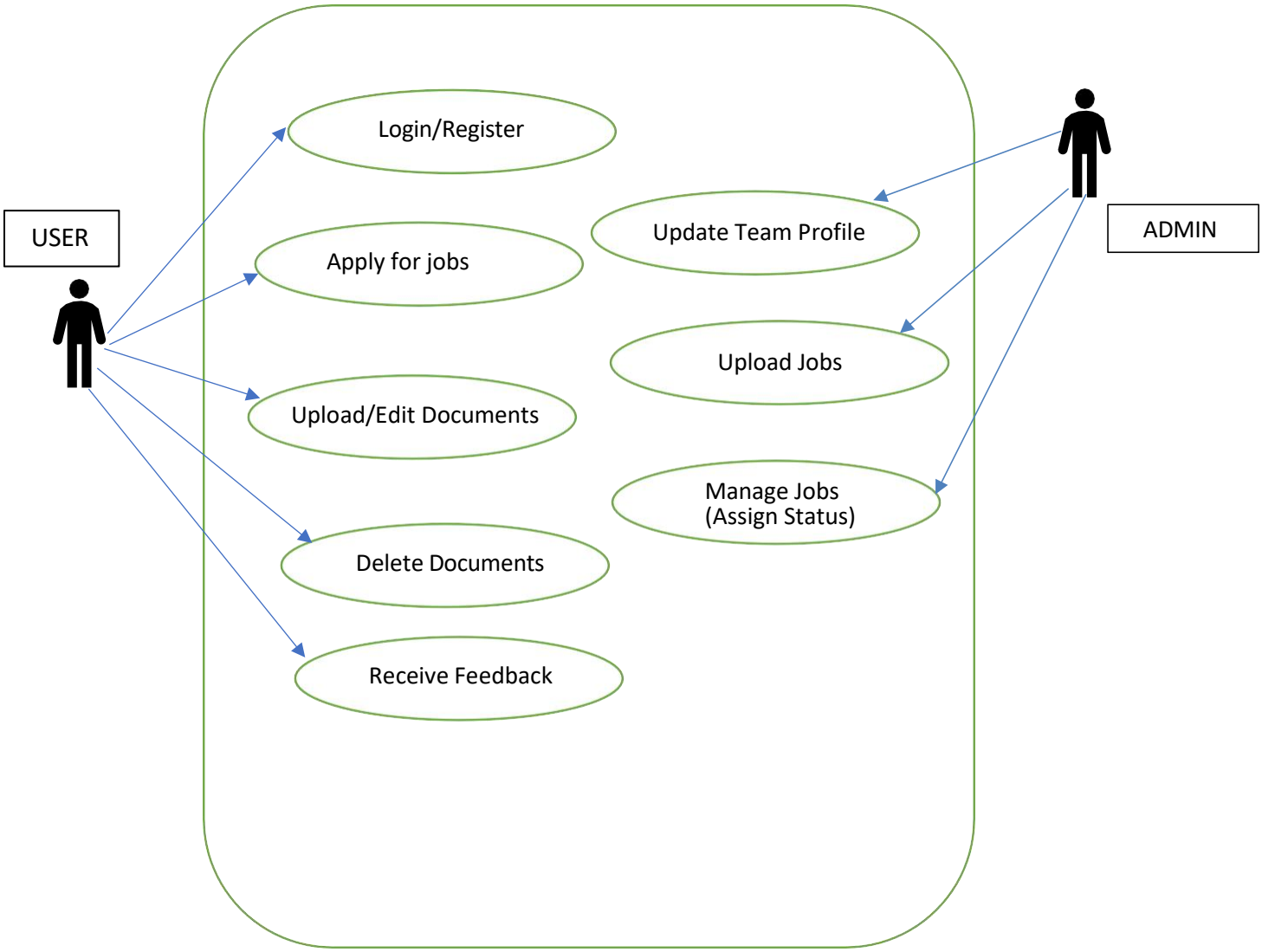
UML DIAGRAMS FOR DOCUMENT MANAGEMENT SYSTEM



Fig, 3 Class Diagram for the Proposed Document Management System

The class diagram reflects key entities and relationships defined in Section 3.4 Formalism. 1-to-1 relations, such as between a user and their role (assign_role: U → R), enforce unique role assignments. 1-to-* relations, like a user to documents or job applications (submit: U × D × F → D, apply_job: U × J × (D ⋁ existing_resume)), represent multiple associations per user. Each class corresponds to formal sets:
Users (U) with credentials and history, Documents (D) with metadata, Forms (F) tied to input and AI suggestions (suggest: F → S), Roles (R) for access control, Applications (A) linking users to jobs and resumes. This visualizes how system components are structurally connected, aligning with their logical mappings.

`



Fig, 4 Use Case Diagram for the Proposed Document Management System

The use case diagram illustrates system interactions between two primary roles from the set of roles $R = \{r_1, r_2, ..., r_t\}$: the User and the Admin (as per assign_role: $U \rightarrow R$). Users (U) can log in (auth: $U \times P \rightarrow \{True, False\}$) defined in Section 3.1, apply for jobs (apply_job: $U \times J \times (D \vee existing\_resume)$) defined in Section 3.5, upload documents (upload: $U \times D \rightarrow F$) defined in Section 2.4, and submit forms (submit: $U \times D \times F \rightarrow D$) enhanced by AI suggestions (suggest: $F \rightarrow S$) defined in Section 3.2 and 3.6 respectively.

Admins manage job listings and user data, reflecting control over job application data ($A = \{a_1, a_2, ..., a_k\}$) and document sets (D), aligning with role-specific permissions.

The diagram visually enforces role-based access boundaries (3.4 Logical Flow – 3.5 Role-Based Access) and maps user interactions to their respective function mappings in the formal model.

`

3.2 PROPOSED ARCHITECTURE

For the proposed Document Management System, a web-based application, the system will follow the Client-Server architecture. Client-Server architecture is a distributed application model that separates tasks or workloads between service providers, known as servers, and service sequester, known as clients. In this setup, when a client makes a request for data via the internet, the server processes the request and sends the requested data back to the client. Clients, in this model, do not share any of their resources. The front-end, which will handle user interactions and displays the interface, will act as the client, while the back-end, secured with TLS, will function as the server, processing requests and managing data storage. The internet will serve as the communication channel between the client and the server.

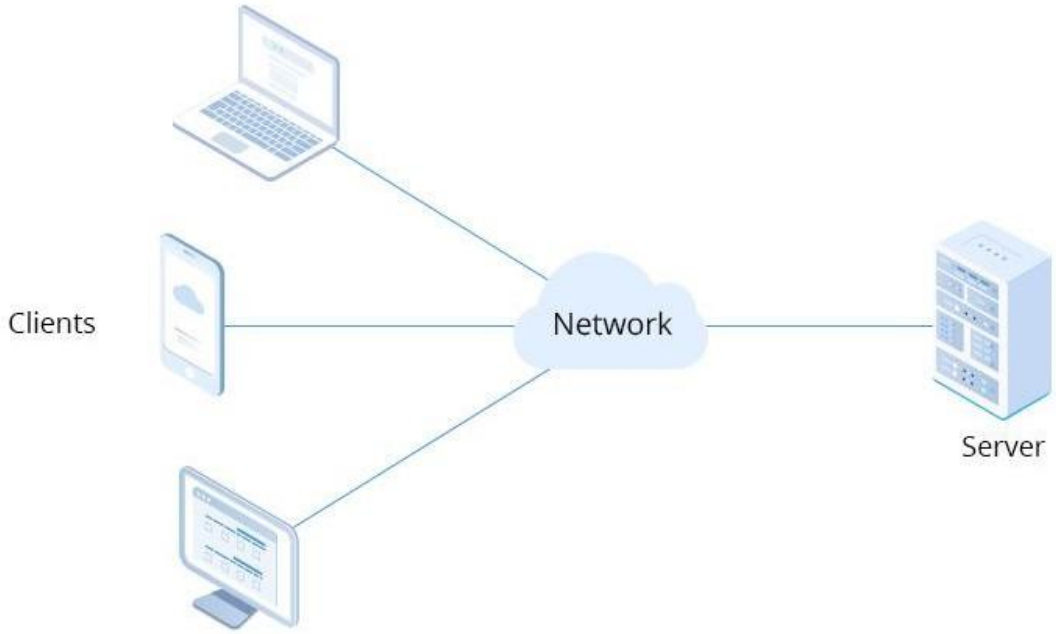


Fig. 5 Proposed Architecture for Document Management System

CLIENT SERVER ARCHITECTURE DIAGRAM

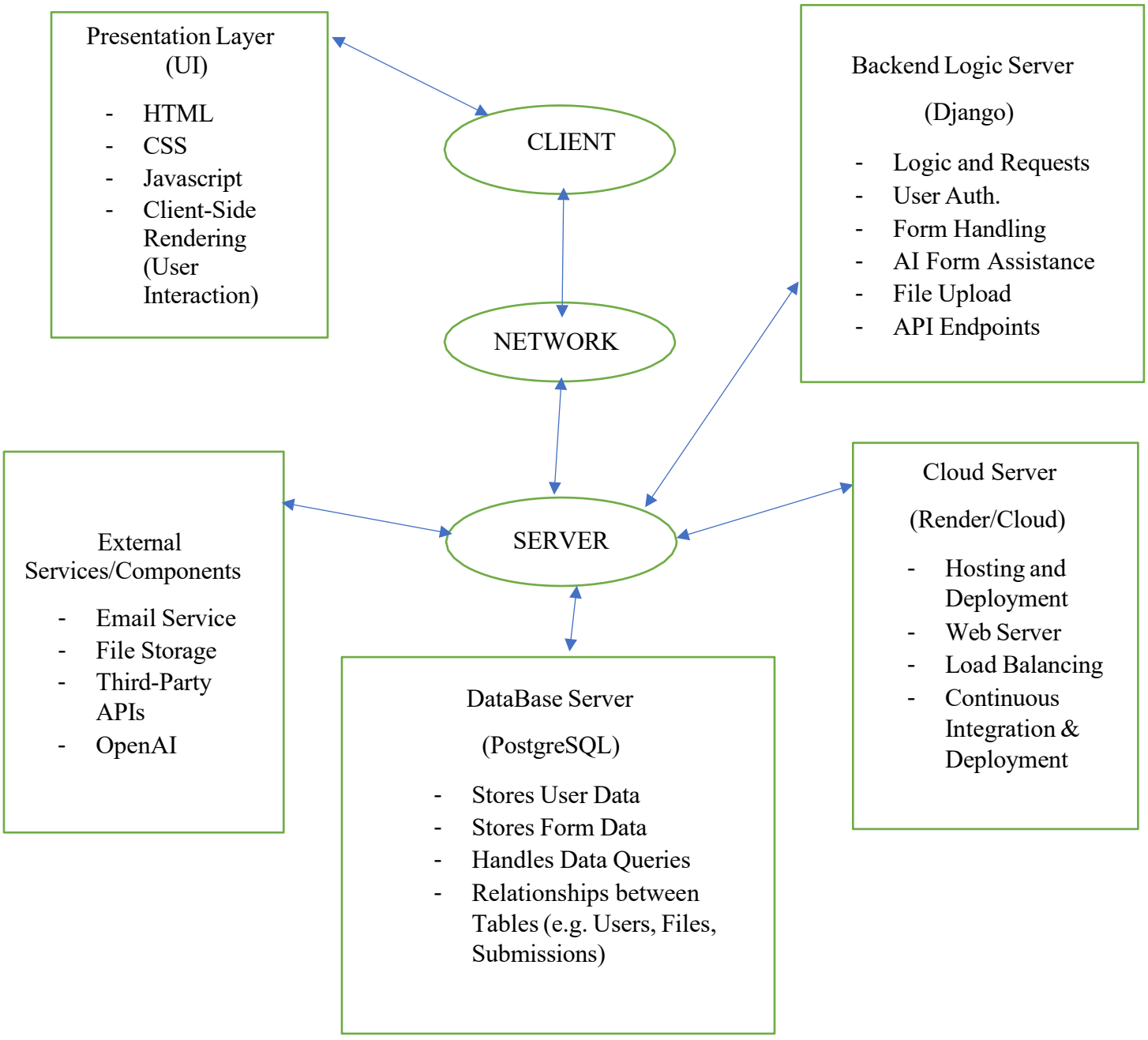


Fig. 6 Proposed Architecture Design for Document Management System

`

1. Presentation Layer (UI Component)

Role: The Presentation Layer is the interface that the user interacts with directly. It will be responsible for rendering the content of the system and facilitating user interactions.

- **HTML**: It will define the structure of the web pages, including the layout of forms, buttons, text fields, and other interactive elements.
    - Dynamic Content: Form fields, error messages, and user-specific content are generated dynamically from the back-end (Django).
- **CSS**: Responsible for styling the UI, ensuring the system is visually appealing and user-friendly.
    - Responsive Design: Ensures the system works on various devices (desktop, mobile, tablet).
    - User Interaction: Interactive elements such as buttons, modals, and pop-up dialog are styled to enhance user experience.
- **JavaScript**: To provide interactivity such as real-time form validation, dynamic content loading, and modal management.

2. Application Layer (Django)

Role: The Application Layer will manage the back-end logic of the system, including user authentication, form processing, business logic, and serving dynamic content.

- Django Framework: It will provide the main framework for the system, handling HTTP requests, routing, views, and interaction with the database.
    - URL Routing: It will determine which view should handle incoming requests (e.g., submission of a form, file download, etc.).
    - Views: Django views will be used for processing requests, including validating data, managing sessions, and returning the appropriate response (HTML page or file).

`

- Templates: Django templates will be used to render dynamic content, filling the HTML structure with real-time data from the back-end (e.g., username, uploaded files).
- Forms and Validation: Django's forms system handles form data, validates it based on specified rules, and ensures that no invalid data is saved.
- AI Form Assistance: AI form assistance helps users by automatically suggesting information in their existing resume from the database when filling application forms, reducing manual effort and improving application speed and accuracy.
- User Authentication: Django handles login, registration, and session management. It ensures users are authenticated before accessing restricted functionality.
- Access Control: It will restrict access to certain data or pages based on user roles (e.g., admin, regular user).
- Error Handling: It will inform users about errors in form submission, document upload, or invalid data input.

3. Data Layer (PostgreSQL)

Role: The Data Layer will store and manage all system data, including user profiles, submitted documents, and metadata. It ensures data integrity, reliability, and security.

- **PostgreSQL Database**: A robust and scalable relational database that stores system data in structured tables.
  - **User Data**: Stores user information (e.g., username, password hash, roles).
  - **Document Data**: Stores metadata related to submitted documents, such as file name, submission date, and associated user.
  - **Form Data**: Stores user-submitted form data for retrieval and future use.
  - **File Storage:** Handles file metadata, including the file path and permissions.

`

- **Relationships:** Establishes relationships between users, documents, and form data to support efficient querying and reporting.
- **Data Integrity:** Ensures that data is stored consistently. This includes enforcing constraints like unique user emails, valid file formats, and correct form inputs.
- **Encryption:** Sensitive data such as passwords are encrypted before being stored to prevent unauthorized access.

4. Infrastructure Layer (Render)

Role: The Infrastructure Layer will be responsible for deploying and hosting the Document Management System, ensuring that the system is available, scalable, and secure.

- **Render (Hosting Platform)**: A platform-as-a-service (PaaS) for hosting Django applications. It manages the application's deployment, scalability, and maintenance.
  - **Web Hosting**: Render ensures that the web application is accessible to users over the internet. It handles the deployment and scaling of the application as needed.
  - **Back-end Processing**: Handles the heavy lifting of rendering dynamic content, processing user requests, and ensuring smooth operation.
  - **Environment Management**: Ensures that the appropriate versions of Python, Django, and other libraries are installed and updated.
  - **Security**: Provides built-in security features, including automatic SSL certificate generation, ensuring encrypted communication between the server and clients.

3.3 SYSTEM ALGORITHM

The system algorithm outlines the core steps involved in the operation of the Document Management System. It provides a structured approach to how the system handles user authentication, form submission, data validation, file uploads, AI suggestions, and overall system workflows.

`

PSEUDOCODE:

```
1. SYSTEM START
    InitializeComponents():
        - Load UI modules
        - Connect to Database
        - Initialize AI Assistance Engine
        - Configure Cloud Storage Integration

2. USER AUTHENTICATION
    -IF NOT IsUserAuthenticated():
        RedirectTo("LoginPage")
    -ELSE:
        LoadDashboard()

3. DOCUMENT / FORM SUBMISSION
    -FOR each page IN FormPages:
        Display(FormFields)

       - IF FormSubmitted():
            -IF NOT ValidateInputs(FormFields):
                DisplayError("Invalid form inputs")
            -ELSE:
                ProcessFormData()
                ai_feedback = RunAIAnalysis(FormFields)
                -IF ai_feedback.HasErrors():
                    SuggestCorrections(ai_feedback)
                SaveToDatabase(FormData)
                UploadToCloudStorage(FormFiles)
```

4. FILE UPLOAD MANAGEMENT

-IF FileUploaded():

-IF NOT ValidateFile(File):

DisplayError("Invalid file type or size")

-ELSE:

UploadToCloudStorage(File)

LinkFileToUser(File, CurrentUser)

5. AI FORM ASSISTANCE

OnFormLoad():

ResumeData = FetchUserResume(CurrentUser)

-IF ResumeData EXISTS:

SuggestedResume = AI_AssistResume(ResumeData)

DisplaySuggestedResume(SuggestedResume)

AllowUserUploadNewResume()

6. JOB APPLICATION PROCESS

job = GetSelectedJob()

-IF UseExistingResume():

resume = FetchUserResume(CurrentUser)

-IF resume IS NULL:

DisplayError("No resume found")

RedirectTo("ResumeUploadPage")

-ELSE:

ProceedWithResume(resume)

-ELSE:

resume = GetUploadedResume()

CreateJobApplication(CurrentUser, job, resume, status="pending")

`

```
    DisplaySuccess("Application submitted")
    RedirectTo("JobListings")

7. DATA ACCESS & AI VALIDATION
    OnDataSubmission():
        result = RunAIValidation(FormData)
        -IF result.HasInconsistencies():
            SuggestCorrections(result)

8. ERROR DETECTION
    -IF SubmissionComplete:
        ai_check = RunAIValidation(SubmittedData)
        -IF ai_check.ErrorsExist():
            SuggestCorrections(ai_check)

9. USER LOGOUT
    -IF LogoutRequested():
        ClearSession()
        RedirectTo("LoginPage")

10. SYSTEM TERMINATION
     TerminateSystem():
         - ReleaseResources()
         - CloseDatabaseConnection()
         - EndSession()
```

3.4 FORMALISM

The formal representation of the system is based on:

1. Set Theory Representation:

1.1 Set of Users (U):

U={u1,u2,...,un} where each user *Uid* has attributes like username, password, role, form submissions, and file uploads.

1.2 Set of Documents (D):

D={d1,d2,...,dm} representing documents with metadata (e.g., file name, upload time, user ID).

1.3 Set of Form Fields (F):

F={f1,f2,...,fp} representing the form fields for document submission.

1.4 Set of Roles (R):

R={r1,r2,...,rt} representing user roles and permissions.

1.5 Set of Job Applications (A):

$A = \{a_1, a_2, ..., a_k\}$ Each application links a user, job, status, and resume version.

2. Function Mapping:

2.1 User Authentication:

auth:U×P→{True,False}, verifying user credentials.

2.2 Form Submission:

submit:U×D×F→D, submitting form data as documents.

2.3 AI Suggestions:

suggest:F→S, providing AI-based suggestions for form fields.

2.4 File Upload:

upload:U×D→F, handling file uploads associated with users.

2.5 Role Assignment:

assign_role:U→R, assigning roles to users.

2.6 Job Application Submission:

apply_job: U × J × (D ⋁ existing_resume)

`

3. Logical Flow of the System:

3.1 User Authentication:

- User logs in, and the system calls auth(ui,p)to validate credentials.

3.2 Form Submission:

- User fillsform, AI calls suggest(fi).
- On submission, system validates and calls submit(u_i, d_j, f_i)

3.3 File Upload:

- On file upload → system invokes upload(u_i, d_j)

3.4 Role-Based Access:

- System assigns roles using assign_role(u_i)

3.5 Job Application Process:

- User selects job → system calls apply_job(u_i, j_j, resume)
- AI retrieves suggested resume if available

3.6 AI Error Detection & Correction:

- After submission → AI validates data consistency and suggests fixes

`

CHAPTER FOUR

IMPLEMENTATION

4.1 INTRODUCTION

This chapter details the implementation process of the Document Management System (DMS), a secure, intelligent solution built to manage document submissions, user records, internal applications, and workforce data. The system was designed using a modular architecture to support secure document storage, user-friendly retrieval, employment tracking, internal job applications, and AI-powered assistance to streamline data entry and error detection during form submission.

4.2 IMPLEMENTATION STEPS

4.2.1 Project Initialization

- Created a Django project structure.
- Set up environment variables for security.
- Installed required packages (cryptography, Pillow, django-cleanup, openAI, django_extensions, Rest Framework API, cloudinary_storage).
- Configured PostgreSQL as the primary database.
- Enabled media and static file handling.

4.2.2 Frontend Development

- Built responsive web interfaces using HTML**,** CSS **(**Tailwind**)**, and JavaScript.
- Developed forms for:
  - Login/Register
  - Personal Data Submission
  - Document uploads
  - Internal job application
- Integrated AI-powered form suggestions that dynamically provide input recommendations and auto-fill options based on users' previously saved data (e.g., personal info, employment history).
- Designed employee profile, workspace, document , job listings and track applications page.
- Designed admin profile, team member and internal jobs dashboard page.

[Screenshot 4.1: Homepage with Navigation Bar and Login/Register Button]

[Screenshot 4.2: User Dashboard Displaying Saved Information and Uploaded Documents]

`

[Screenshot 4.3: Job Application Form with AI Suggestion Dropdowns]

4.2.3 Backend Development

- Developed models for:
  - CustomUser(AbstractUser)
  - UserProfile
  - CompanyProfile
  - Documents
  - JobApplication
  - CompanyTeamMember
- Created secure file upload logic with:
  - File type validation
  - Size constraint
  - Encrypted file storage
- Implemented AI Integration
  - Backend APIs connect with AI modules to analyze submitted forms in real-time.
  - AI offers personalized auto-fill suggestions from user profile data to reduce repetitive typing.
- Integrate authentication and authorization for access-level controls(e.g admin vs employee).
- Developed reusable components and serializers for interactions between frontend and backend using Django REST Framework.

4.2.4 Core Features Implemented

- Document Submission Module: Users can upload resumes, certifications, and other professional documents. Each file is versioned, encrypted, and linked to their profile.
- AI-Powered Form Assistance:
  - When users access or resume forms, the system retrieves existing data and uses AI algorithms to auto-suggest or pre-fill fields.
  - AI detects common errors (e.g., missing mandatory fields, invalid date formats) and prompts users with corrective suggestions before submission.
- Data Encryption: Sensitive data (e.g., document contents) is encrypted using the cryptography library before storage.
- Profile Management: Users can update and maintain employment history and personal details.
- Employee Management Module: Admin users can view, edit, and manage staff profiles, including job roles and application activity.
- Job Application System: Enables employees to apply for internal/external job openings directly within the platform.

`

- Real-Time Job Application Tracking: Enables employees to monitor the status of their job applications instantly, from submission to decision.
- Personalized Employee Workspace for Portfolio: Provides a personalized space to manage job history, documents, and build a shareable digital portfolio.
- Document Tracking Page: Centralized dashboard where users can view and manage all submitted personal and official documents.

[Screenshot 4.4: Admin View of Employee Records]

[Screenshot 4.5: File Upload Modal with Validation and AI Form Suggestions]

4.2.5 Testing and Validation

- Conducted:
  - Unit testing for views, models, utitlity function and forms.
  - Manual authentication check for user login.
  - Manual validation of document encryption/decryption flow.
  - Role-based access control testing (admin, staff, applicant).
  - AI form assistance accuracy testing, ensuring suggestions are relevant and improve user experience.
- Tested scenarios include:
  - File type rejection (e.g., .exe, .bat)
  - Template rendering accuracy for user profiles
  - Data integrity with saving files/documents
  - Real time job application tracking
  - AI suggestion acceptance and error detection in job application forms.
-

4.2.6 Deployment

- Conducted:
  - Deployed using Render for the web application and Cloudinary for secure file storage.
  - Configured PostgreSQL with persistent storage.
  - Enabled HTTPS and CSRF protection for user data security.
  - Integrated AI services in production environment with optimized latency.

[Screenshot 4.6: Live Application Environment After Deployment]

`

4.3 TOOLS AND TECHNOLOGIES USED

Component	Technology Used
Frontend	HTML, CSS (Tailwind), JavaScript
Backend	Django, Django REST Framework
Database	PostgreSQL
File Handling	Cloudinary, Pillow, django-cleanup
Encryption	Cryptography Library (Fernet)
AI Integration	OpenAI modules for form suggestions and error detection
Deployment	Render(app),Cloudinary (files)

4.4 SUMMARY OF KEY FEATURES

Module	Feature Summary
User Management	Profile creation, login/logout, data update
Document Management	Upload, validate, encrypt, retrieve, version documents
Employment History	Add/edit/view work records
Job Application	Submit internal/external applications using AI-enhanced forms
AI Form Assistance	Personalized auto-fill, input validation, error correction
Admin Dashboard	View, filter, and manage employee records and submissions

4.5 CHALLENGES AND MITIGATIONS

Challenge	Resolution Strategy
Secure file encryption	Used Fernet encryption with secret key rotation
Resume versioning complexity	Designed version model with timestamp tracking
Role-based access handling	Implemented Django user groups and permissions
File upload validation issues	Added client- and server-side checks for file types
AI Suggestion Accuracy	Iteratively refined AI models; incorporated user feedback

`

4.6 CONCLUSION

The implementation of the Document Management System successfully integrated AI-powered form assistance, enhancing user experience by reducing errors and repetitive data entry. Coupled with secure document handling, encrypted storage, and a robust job application module, the system streamlines organizational workflows and employee management.

`

CHAPTER FIVE

EXPECTED CONTRIBUTION TO KNOWLEDGE

5.1 INTRODUCTION

This chapter presents the anticipated contributions of the developed Document Management System (DMS) to both academic research and real-world applications. It emphasizes how the system enhances document workflows, reinforces data protection, and introduces intelligent features that promote operational efficiency. Furthermore, it introduces the innovative use of the system as a dynamic portfolio platform for individuals to showcase their job history and professional credentials.

5.2 CONTRIBUTION TO KNOWLEDGE

1. AI-Enhanced Form Interaction and Smart Suggestions
The system integrates AI-driven form suggestions to improve the accuracy and efficiency of form completion. It intelligently pre-fills data based on previously stored user records, highlights potential input errors, and provides correction hints, paving the way for a more user-friendly and intelligent document management experience.

2. Secure Document Lifecycle Management
Through structured validation, encryption protocols, and cloud-based storage mechanisms, the system presents a robust model for secure document handling. This ensures data integrity, version control, and reliable access management while maintaining user privacy and compliance with data protection standards.

3. Unified Workforce and Document Management Architecture
The DMS serves as a dual-purpose tool by integrating employee management functionalities alongside traditional document storage. It allows for centralized tracking of team members' professional records, document submissions, and employment milestones, supporting administrative operations such as internal job postings and employee transfers.

4. Scalable and Deployable Enterprise Solution
With its use of Django, REST APIs, and modern deployment platforms (Heroku, AWS), the project provides a deployment-ready framework. This makes it valuable not only academically but also as a reference for tech startups and enterprise software teams looking to implement efficient internal documentation systems.

5. Promotion of AI Adoption in Administrative Systems
The DMS sets a precedent for applying AI in operational domains like HR, recruitment, and compliance

`

documentation. By including intelligent features in everyday admin tasks, the system demonstrates how AI can move beyond analytics into process automation and intelligent assistance.

6. Reusability and Intelligent Application Support

The system enhances usability by allowing users to reuse saved data during future form submissions or job applications. This significantly reduces repetitive data entry, minimizes human error, and promotes consistency in user information across various organizational processes.

7. Personal Portfolio and Career Management Platform

One of the innovative aspects of this research is enabling the system to act as a professional portfolio and resume hub. Users can save and maintain their:

- Personal details
- Educational and employment history
- Certifications and achievements
- Uploaded documents

5.3 CONCLUSION

This research introduces a multifunctional Document Management System that not only addresses operational needs such as secure document submission, employee tracking, and AI-assisted form handling, but also empowers users with a platform to manage and showcase their professional history. It contributes practical value to organizations and individuals alike, serving as a blueprint for intelligent digital administration and personal career management.

REFERENCE

Sharma, A., & Patel, S. (2020). Building Document Management Systems: A Guide to Best Practices. Springer. Wang, L., & Zhang, Y. (2019).

Intelligent Form Field Assistance Using Machine Learning. Journal of AI & Data Science, 15(3), 88-95. Meyer, R., & Gupta, K. (2020).

Building Document Management Systems: A Guide to Best Practices, Adeyemi and Thomas (2020).

Data Security in Document Management Systems: Encryption and Validation. Wiley. Mohammad, S., & Lee, C. (2018).

The Role of AI in Enhancing User Experience in Web Applications. AI and Web Technologies Journal, 22(4), 102-110. Soni, P., & Jain, M. (2017).

AI Algorithms for Web Form Assistance. Proceedings of the 2nd International Conference on Data Science and Machine Learning, 215-223. Fitzgerald, S., & Patel, R. (2021).

Designing Scalable and Secure Web Applications. Pearson. Django Software Foundation. (2024).

Django Documentation - Introduction to Django and its features. Evans, T., & Green, R. (2019).

Leveraging cloud technologies in document management systems. Journal of Cloud Computing, 12(3), 23-34. Jackson, R., & Lee, A. (2018).

Database technologies for document management systems: A comparative analysis. Database Systems Review, 25(2), 89-104. Smith, L. (2020).

Developing secure document management systems using Django. Web Application Development Journal, 18(4), 45-59.

`