

**THE STATE UNIVERSITY OF ZANZIBAR SCHOOL OF BUSINESS (SOB)**

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**FINAL PROJECT REPORT**

**PROJECT TITTLE: UNIDOCS – DOCCUMENTS MANAGEMENT SYSTEM FOR UNIVERSITIES.**

**STUDENT NAME: RAMADHANI YASSIN RAMADHANI**

**REGISTRATION NO: BITA/6/22/079/TZ**

**DEGREE COURSE: ICT WITH ACCOUNTING**

**SUPERVISOR: MR. HASSAN MOHAMMED**

# **Declaration**

I confirm that this report is my own original work. To the best of my knowledge, it does not include material from other sources unless properly cited. No part of this work has been submitted for any other degree or diploma at any institution.

**Signature: Date:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 10/08/2025

# **Abstract**

UNIDOCS - Integrated Document Management System for Universities is built to simplify the process of creating, handling, and managing official documents in higher learning institutions. The existing manual system is slow, prone to mistakes, and inefficient, causing unnecessary delays. UNIDOCS incorporates a chatbot for instant assistance and a structured document library for better access and management. This system automates document requests, keeps track of progress, and sends notifications, making processes smoother and more transparent. Built with Angular for the frontend, Spring Boot for the backend, and Postgres for data storage, UNIDOCS aims to make university administration more efficient, secure, and accessible.   
The System also encorporates the CV generation modules and Announcements Management.  
This report covers the purpose, challenges, and methods used in developing the system, focusing on improving administrative workflows in universities.

# **Dedication**

I dedicate this work to my family, friends, and mentors, who have continuously supported and motivated me. Their encouragement has been my driving force throughout this journey.

# **Acknowledgement**

I sincerely thank my supervisor for his valuable guidance and support. I also appreciate my colleagues and friends for their encouragement. A special thank you to the university Administration and students for sharing their insights, which have played a crucial role in shaping UNIDOCS into a practical solution for document management.

**Table of contents**

[Declaration i](#_Toc1706824432)

[Abstract ii](#_Toc1801908489)

[Dedication iii](#_Toc1937014498)

[Acknowledgement iv](#_Toc1966766419)

[Table of contents v](#_Toc327198090)

[Table of Figures vii](#_Toc1931933690)

[CHAPTER 1: INTRODUCTION 1](#_Toc79200876)

[1.1 Introduction 1](#_Toc110346716)

[1.2 Project Background and Motivation 1](#_Toc551086917)

[1.3 Problem Statement 2](#_Toc1220216481)

[1.4 Problems Solution and Scope 2](#_Toc117994789)

[1.5 Feasibility Study 3](#_Toc986995786)

[1.6 Conclusion 4](#_Toc150637551)

[CHAPTER 2: LITERATURE REVIEW 5](#_Toc1040418061)

[2.1 Introduction 5](#_Toc191581154)

[2.2 Related Work 5](#_Toc332052624)

[2.3 Previous Systems 6](#_Toc1681290010)

[2.4 Lessons Learned. 10](#_Toc1263590133)

[2.5 Critique of the Review 10](#_Toc12987117)

[2.6 Conclusion 10](#_Toc1078852537)

[CHAPTER 3: PROJECT METHODOLOGY 11](#_Toc1491907833)

[3.1 Introduction 11](#_Toc320252730)

[3.2 Software Development Approach 11](#_Toc1077206424)

[3.3 Software Development Life Cycle Model (SDLC) 12](#_Toc488088851)

[3.4 Software Development Tools 13](#_Toc2040541444)

[3.5 Information Gathering and Analysis 14](#_Toc1982917231)

[3.6. System Analysis 15](#_Toc1652103905)

[3.6.1. Modeling Approach Selection 15](#_Toc830796868)

[3.6.2. Modeling Tools Used 16](#_Toc162713715)

[3.6.3. Key System Components 16](#_Toc1063163109)

[CHAPTER 4: SYSTEM ANALYSIS 18](#_Toc390137653)

[4.1 Existing System 18](#_Toc1964622204)

[4.1.1 Existing System Description 18](#_Toc852693959)

[4.1.2 Business Rules 19](#_Toc209420424)

[4.2 Requirements Specification 19](#_Toc144336646)

[4.2.1 Functional Requirements 19](#_Toc637144002)

[4.2.2 Non-functional Requirements 20](#_Toc2040781120)

[4.2.3 Performance Requirements 20](#_Toc223537522)

[4.2.4 Software and Hardware Requirements 20](#_Toc747490718)

[4.2.5 Preliminary Product Description 21](#_Toc444384390)

[4.3. System Modeling 22](#_Toc1443754004)

[4.4. Dataflow Diagram 22](#_Toc865485508)

[4.5 Requirement Structuring 23](#_Toc1431380176)

[4.5.1 Use case diagram 23](#_Toc1594391555)

[4.5.2. Use Case Documentation 24](#_Toc1905903569)

[4.5.4. Conceptual modelling: Class diagram 26](#_Toc1622961330)

[4.5.5. Entity Relationship diagram 27](#_Toc1926444179)

[References 28](#_Toc1439709931)

# **Table of Figures**

[Figure 1 : EduSec 7](#_Toc922423272)

[Figure 2 : DocuWave 8](#_Toc181415073)

[Figure 3 : M-Files 9](#_Toc1072008979)

[Figure 4 : Agile Methodology 12](#_Toc1545046175)

[Figure 5 : Dataflow Diagram 22](#_Toc307265613)

[Figure 6 : Use case diagram 23](#_Toc1143664666)

[Figure 7 :Sequence Diagram 25](#_Toc905710807)

*[Figure](#_Toc937739072)* [8](#_Toc937739072) *[: Class diagram](#_Toc937739072)* [26](#_Toc937739072)

[Figure 9 : Entity Relationship diagram 27](#_Toc1558542852)

[Table 1 : Core Entities 16](#_Toc1141015605)

[Table 2 : Supporting Services 17](#_Toc435908868)

# **CHAPTER 1: INTRODUCTION**

## **1.1 Introduction**

UNIDOCS - Integrated Documents Management System for Universities is designed to simplify the way universities manage and process official documents. The current system in most institutions is largely manual, leading to inefficiencies, long processing times, and increased chances of human errors. Students and staff often struggle with delays in obtaining important documents, such as transcripts, recommendation letters, and administrative approvals.  
By implementing an advanced digital platform, UNIDOCS will automate document processing, reduce administrative burdens, and improve accessibility. The System has also a CV management portal where by Students wll be molding their Resumees for Career development. In other ways the System seamlesly allows automated Emailing of important Letter Copies, Public Announcements and Updates on the Requested Letter whenever they got approved, Rejected or Kept Pending. The UNIDOCS System also feature an AI-powered chatbot to assist users with inquiries. With a user-friendly design, UNIDOCS will enhance efficiency, transparency, and accuracy in university document management.

## **1.2 Project Background and Motivation**

Universities handle thousands of documents every academic year. Traditional document management systems rely on physical paperwork, making the process time-consuming and prone to misplacement or delays. Staff members often find it difficult to track document requests, and students experience frustration when following up on approvals.  
The increasing digital transformation in education has highlighted the need for modern, automated solutions to streamline administrative processes. Many sectors have adopted AI-driven tools to optimize operations, but university document management remains largely outdated. UNIDOCS is motivated by the need to bridge this gap by providing an intelligent, automated, and secure document management system tailored for university environments.

## **1.3 Problem Statement**

The current document handling system in universities presents several challenges:

1. Heavy reliance on manual paperwork, causing delays and inefficiencies.
2. Lack of a centralized platform for tracking and managing document requests.
3. Difficulty in retrieving and verifying documents due to disorganized storage.
4. Absence of an automated system for answering student and staff FAQ about document processing.
5. Abense of unified CV Management portal amongst Students
6. Lack of efficient Updates and Announcement Management on the University.

These issues create administrative bottlenecks, leading to dissatisfaction among students and staff. There is a clear need for a digital solution that enhances document processing, storage, and retrieval while providing instant support through AI-driven assistance.

## **1.4 Problems Solution and Scope**

UNIDOCS aims to solve these problems by developing an Integrated Documents Management System with the following key features:

1. **Automated Document Processing**: Streamlines request submissions, approvals, and retrievals.
2. **AI-Powered Chatbot**: Provides real-time assistance to students and staff regarding document-related inquiries.
3. **Notification and Tracking System**: Keeps users updated on the status of their document requests.
4. **CV Management Modules**: THe System has an interface for Generating Students Resumees for Their career path.
5. **Announcements and Updates**: The system Admins will be capable to publish Announcements and automatically got emailed to all Students.

The scope of this project includes system design, development, testing, and deployment, ensuring a scalable and user-friendly platform for university administration.

## 

**General Objective:**   
The Main objective of this project is to modernize University document management by developing an automated, AI-driven platform that enhances efficiency, accessibility, and accuracy.

**Specific Objectives:**

1. Implement a document request and tracking system to streamline university administrative processes.
2. Structure an AI-powered BOT using chatbase.co API for Streamed Answering FAQ.
3. Implementing Resumee / CV generation Modules using pre set well structured CV Templates on the System.
4. To structure announcements and Updates Management portal and Automated Emailing and Notifications to Students.

## **1.5 Feasibility Study**

**Economic Feasibility**: The project utilizes open-source technologies (Angular, Spring Boot and Postgres) to minimize costs while maintaining efficiency.  
**Technical Feasibility**: The System utilizes the use of Modern web technologies (Angular and Spring boot), and Flutter for Mobile app ensuring the successful implementation of the system.  
**Legal Feasibility**: The UNIDOCS System do not violate legal laws and Regulations as it maintains privacy and Good conduct no immoral acts are potrayed on overall system processes.  
**Operational Feasibility**: UNIDOCS addresses real administrative challenges faced by universities, making it highly practical and beneficial for users.

## **1.6 Conclusion**

By implementing UNIDOCS, universities will transition from slow, manual processes to an intelligent, automated document management system, ultimately enhancing productivity and user satisfaction.

# **CHAPTER 2: LITERATURE REVIEW**

## **2.1 Introduction**

This chapter provides an in-depth review of existing literature on document management systems, particularly in university settings. By analyzing past research, related projects, and existing systems, we aim to identify gaps and establish a foundation for UNIDOCS. The review focuses on automated document management, AI-powered chatbots for administrative support, and best practices in digital record-keeping. This chapter is divided into five sections: Related Work, Previous Systems, Lessons Learned, Critique of the Review, and Conclusion.

## **2.2 Related Work**

Universities and educational institutions worldwide rely on document management systems to handle administrative tasks efficiently. Various platforms exist to assist in managing student records, faculty documentation, and official communications. However, many of these systems lack real-time interaction features, automation, or integration with AI-driven assistance.

Research indicates that document management solutions enhance efficiency by reducing paperwork and streamlining workflows (Smith et al., 2021). AI-based systems, such as chatbots, have also proven effective in reducing administrative workloads by handling frequently asked questions and guiding users through processes (Jones & Patel, 2020). However, existing university-based document management systems still suffer from fragmented services and poor user experience.

Some widely used document management systems in universities include:

* **DocuWare** – A cloud-based system for digital document storage and workflow automation.
* **M-Files** – Focuses on metadata-driven organization but lacks AI integration for student support.
* **Ellucian Banner** – A comprehensive ERP for university administration but complex and expensive for many institutions.
* **EduSec** – A student information system that includes document management features, though it lacks AI-driven automation and chatbot integration.

These platforms have improved document management but still leave gaps in accessibility, automation, and user interaction, which UNIDOCS aims to address.

## **2.3 Previous Systems**

Several document management systems have been developed to help universities streamline administrative processes, manage student records, and improve workflow efficiency. These systems aim to reduce paperwork, enhance document security, and provide seamless access to critical academic and administrative documents. While some platforms focus on simple document storage, others incorporate advanced features like automation, AI assistance, and integration with university databases. The following platforms represent significant advancements in university document management:

**2.3.1 EduSec**

* **URL**: <https://www.edusec.org>
* **Platform**: Web-based

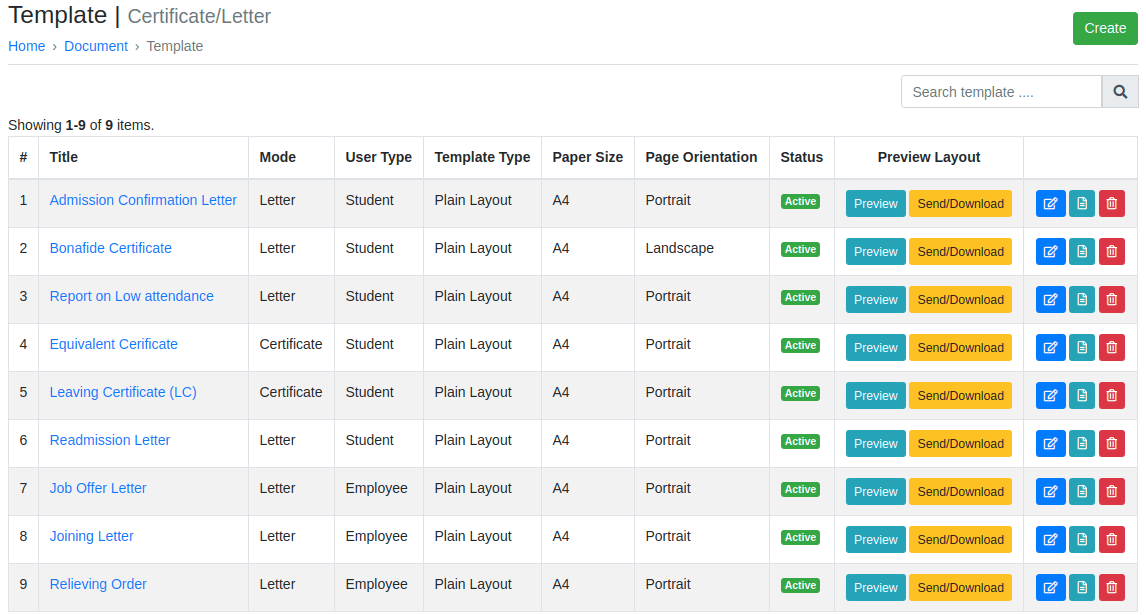


Figure 1: EduSec

**EduSec** is a student information system that incorporates basic document management features. It provides tools for managing academic records and administrative documents. However, it lacks automation and AI-powered functionalities, which limits its ability to enhance efficiency in document processing.

**2.3.2 DocuWare**

* **URL**: <https://www.docuware.com>
* **Platform**: Web-based

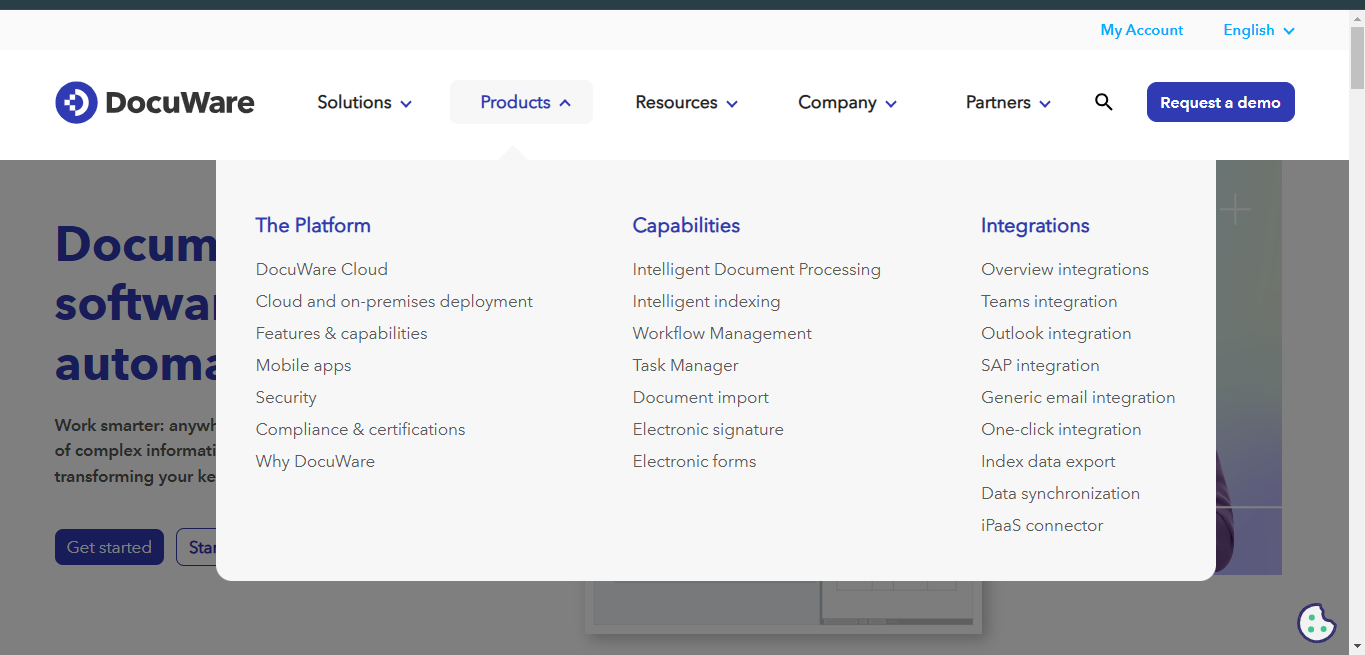


Figure 2: DocuWave

**DocuWare** is a cloud-based document management system that offers workflow automation and digital storage solutions. It enables universities to digitize paper-based processes, improving accessibility and reducing human errors. However, DocuWare lacks AI-driven assistance, making user interactions more manual and less responsive.

**2.3.3 M-Files**

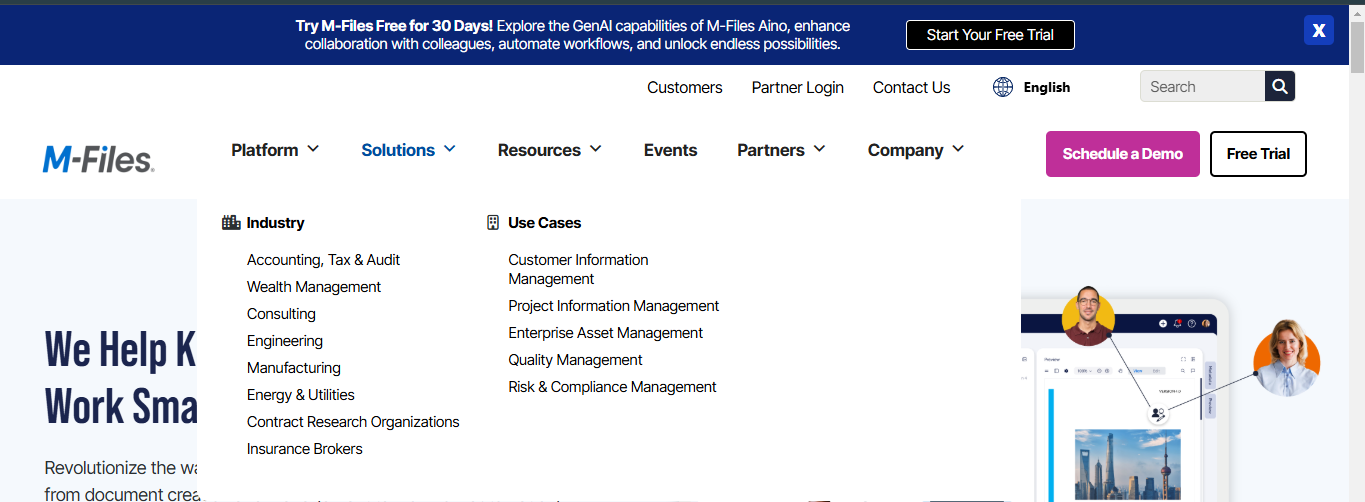
* **URL**: <https://www.m-files.com>
* **Platform**: Web and Mobile

Figure 3: M-Files

**M-Files** is another widely used system that organizes documents based on metadata rather than traditional folder structures. This approach simplifies document retrieval and reduces the chances of misplacement. However, M-Files does not provide real-time AI chatbot support, which could assist students and staff in document-related queries.

**2.4 Lessons Learned.**From reviewing existing literature and systems, we can conclude the following:

1. **Need for Automation** – Many current university document management systems still require manual processes, leading to inefficiencies.
2. **Integration of AI Chatbots** – AI-powered chatbots can enhance user experience by providing instant assistance for document-related queries.
3. **Accessibility and User-Friendly Design** – Most platforms lack intuitive user interfaces, making navigation difficult for students and staff.
4. **Scalability** – Many existing systems are rigid and do not easily adapt to growing user demands.

## **2.5 Critique of the Review**

While existing document management solutions improve efficiency, they often lack essential features such as AI-driven support, automation, and a centralized platform for all administrative documents. Many systems focus solely on document storage but fail to integrate tools that enhance user engagement and self-service capabilities.  
Additionally, security and compliance remain concerns in many systems, as universities deal with sensitive student and faculty information. The need for enhanced access control and verification mechanisms is essential for a secure and trustworthy document management system.

## **2.6 Conclusion**

This literature review highlights the need for an integrated, AI-powered document management system tailored for universities. Existing systems offer partial solutions but fail to address automation, real-time support, and user accessibility. UNIDOCS seeks to bridge these gaps by providing a scalable, efficient, and intelligent platform for university document handling, ensuring a seamless experience for students and staff alike.

# **CHAPTER 3: PROJECT METHODOLOGY**

## **3.1 Introduction**

This chapter details the methodology used in the development of the UNIDOCS - Integrated Document Management System for Universities. The chosen approach ensures a systematic and structured process for system development, covering information gathering, system analysis, requirements specification, and design choices. This methodology helps in delivering a scalable, efficient, and user-friendly solution for document automation and management within university environments.

## **3.2 Software Development Approach**

The development of UNIDOCS follows an **Object-Oriented Approach (OOA)** to enhance modularity, maintainability, and scalability. The system is structured into independent yet interconnected modules, including **User Management, Document Processing, AI Chatbot, Request Tracking, and Notifications**. Each module operates autonomously while seamlessly integrating through Restful APIs and a well-structured Postgres database for efficient data storage and retrieval.

A **Bottom-Up Approach** is being adopted, where individual components such as classes and objects were developed first and later integrated into a fully functional system. Unified Modeling Language (UML) diagrams, including **Use Case, Class, and Sequence Diagrams**, were utilized to visualize system interactions and structure. The AI-powered chatbot, implemented using  **PAWA API and Mocking-Pawa Inline Saved Codes**, enhances user experience by providing real-time document-related assistance. This structured methodology ensures that UNIDOCS remains flexible, scalable, and ready for future enhancements while maintaining high performance, security, and reliability.

## **3.3 Software Development Life Cycle Model (SDLC)**

The **Agile Model** is being selected for this project due to its iterative and flexible nature, allowing for continuous user feedback and frequent improvements. The key phases of the development process include:

1. **Requirement Analysis**: Gathering functional and non-functional system requirements through surveys, interviews, and literature reviews.
2. **Design**: Developing system architecture diagrams and defining data flows to ensure a seamless structure.
3. **Implementation**: Developing the frontend using **Angular** and the backend with **Spring Boot**, ensuring system functionality as planned.
4. **Testing**: Conducting **unit, integration, and system testing** to identify and resolve issues early in the development cycle.
5. **Deployment & Maintenance**: Deploying the system and providing ongoing updates to maintain performance, security, and scalability.

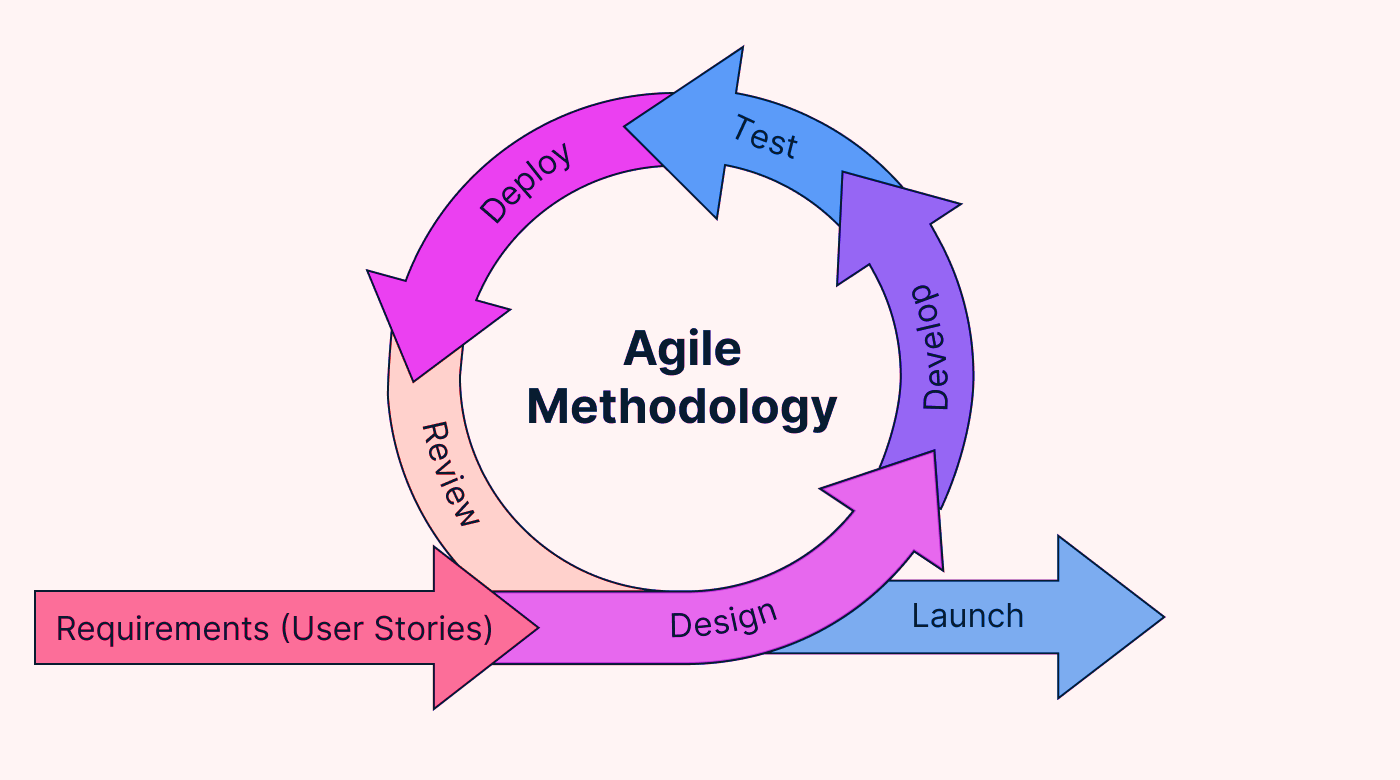


Figure 4: Agile Methodology

## **3.4 Software Development Tools**

To ensure the successful completion of UNIDOCS, the following software tools were employed during different phases:

**Design:   
**Tools**: Figma.**  
**Purpose**: Figma is being used for wireframe design of the user interface, helps to visualize layouts, User Flows and User Interfece before Actual Development.

**Implementation:**

1. **Frontend Development**: Angular and Flutter (For Mobile App)
2. **Backend Development**: Spring Boot, Java
3. **Database**: PostgreSQL
4. **Version Control**: Git, GitHub

**Purpose**: Angular is being chosen for a dynamic and interactive frontend, while Spring Boot is being used for developing robust backend services. GitHub facilitated collaboration and version control.

**Testing:   
**Integration Testing**: Postman**   
**Purpose**: Used for testing API endpoints by sending requests and verifying responses, ensuring that different software components integrate and function correctly together.

**System Development Platform:**

UNIDOCS is being developed using a combination of software tools, programming environments, and hardware units to ensure **optimal performance and scalability.**  
**Hardware Units:  
Development Machines**: Standard workstations with high CPU and RAM configurations for coding, testing, and running simulations.

**Programming Environment:**

1. **Frontend**: Angular.js and Flutter for creating dynamic and responsive user interfaces.
2. **Backend**: Spring Boot for developing scalable Restful APIs.
3. **Database**: Postgres for structured data management.
4. **Version Control**: Git and GitHub for source code management.
5. **Containerization**: Docker for ensuring a consistent environment across development, testing, and production.

## **3.5 Information Gathering and Analysis**

To gain a comprehensive understanding of the problem, multiple data collection techniques were employed:

1. **Interviews**: Conducted with university staff and students to understand document management challenges.
2. **Literature Review**: Examined best practices and previous research on document automation.
3. **Questionnaires**: Surveys were distributed to collect feedback on system functionality and ease of use.

## **3.6. System Analysis**

System analysis is a systematic approach that uses graphical tools to analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by users.

### **3.6.1. Modeling Approach Selection**

After requirement gathering, the system requirements were structured and software models were developed using the **Object-Oriented Approach** combined with **Bottom-Up Development Strategy**.

**Why Object-Oriented Approach was Chosen:**

* **Clear Object Boundaries**: The system has well-defined entities (User, LetterRequest, CVRequest, Announcement) with distinct properties and behaviors that naturally map to objects.
* **Inheritance and Polymorphism**: The system utilizes different user roles (admin/student) and various request types that can be effectively modeled through inheritance hierarchies and polymorphic behavior.
* **Encapsulation**: Each entity encapsulates its data and related operations, with services handling business logic and repositories managing data access.
* **Real-world Modeling**: The system directly models real-world academic processes (letter requests, CV generation, announcements) making object-oriented design intuitive.
* **Reusability**: Services and components can be reused across different parts of the system, promoting code efficiency.
* **Maintainability**: Object-oriented principles provide better code organization and easier maintenance.

### **3.6.2. Modeling Tools Used**

**Object-Oriented Modeling Tools:**

* **Class Diagrams**: Depict the system's classes, their attributes, methods, and relationships
* **Use Case Diagrams**: Show system functionality from user perspective
* **Sequence Diagrams**: Illustrate object interactions and message flow
* **State Chart Diagrams**: Model object state transitions
* **Activity Diagrams**: Represent business processes and workflows

### **3.6.3. Key System Components**

**Core Entities:**

The UNIDOCS comprises of various Core Entities including the User, LetterRequest, CVRequest and Announcement.

|  |  |
| --- | --- |
| **Entity** | **Description** |
| **User** | Represents system users (Students and Admins) |
| **LetterRequest** | Manages academic Letter Applications |
| **CVRequest** | Handles CV generation requests |
| **Announcement** | Manages system announcements and notifications |

Table 1: Core Entities

**Supporting Services:**

The UNIDOCS System comprises of Multiple supporting services to Ensure the streamlined Functionality of Pre described Features. The Table Below Highlits the UNIDOCS Supporting Services:

|  |  |
| --- | --- |
| **Supporting Service** | **Explanations** |
| **EmailService** | Handles email notifications and password resets |
| **JwtService** | Manages authentication and authorization |
| **PasswordResetService** | Handles password recovery processes |
| **PawaAIService** | AI-powered CV and letter generation guidance |

Table 2: Supporting Services

**Frontend Applications:**

**Angular Web Application**: Rich web interface for both admin and student portals.

**Flutter Mobile Application**: Authentic mobile experience for Students.

The combination of Object-Oriented Analysis with Bottom-Up development strategy ensures that the UNIDOCS system is not only well-designed but also reliably implemented with proper testing and validation at each development stage.

# **CHAPTER 4: SYSTEM ANALYSIS**

## **4.1 Existing System**

The current system for document management in universities is highly reliant on manual processes, which results in inefficiencies and delays. Document requests, approvals, and storage are handled through physical paperwork or scattered digital files, leading to difficulties in tracking and retrieval. Students and staff must visit administrative offices in person to request official documents, such as transcripts and recommendation letters, which can take days or even weeks to process.

### **4.1.1 Existing System Description**

.  
Data sharing and Communication between students and administrative personnel is slow and unstructured, oftenly relying on physical visits. The absence of a **centralized digital platform** causes delays in processing requests and creates a high workload for administrative staff. There is no **automated verification system**, making it challenging to confirm document authenticity and track request statuses in real time.

**Key limitations of the existing system include:**

1. **Manual document handling:** Requests and approvals are paper-based, increasing processing time and risk of document misplacement.
2. **Lack of a centralized system:** Students, staff, and administrators do not have a unified platform to manage document requests and approvals efficiently.
3. **Limited automation:** Processes such as document verification, request tracking, and notifications are performed manually.
4. **Slow communication:** Students must follow up on requests in person or through email, often leading to miscommunication and delays.

### **4.1.2 Business Rules**

The current document management system follows these operational rules:

1. Students must submit paper-based forms or emails to request official documents.
2. Document requests are reviewed manually by administrators before approval.
3. Communication about document status is done through email or physical notice boards.
4. Approved documents are stored physically or in unstructured digital folders without a centralized retrieval system.
5. Any modification or update to a document requires manual review and approval by the administrative office.

**4.2 Requirements Specification**

Requirements Specification is a crucial phase in software development that involves documenting the detailed functional and non-functional requirements of the system to be developed.

**4.2.1 Functional Requirements**

The proposed **UNIDOCS system** aims to address the challenges of the current system by incorporating the following key functionalities:

1. Students can submit document requests through an online portal.
2. Administrators can review, approve, or reject requests digitally.
3. The system Securely encrypts user confidential details like Passords for ensuring document authenticity and Confidentiality.
4. The System automatically sends Email to Users (Students) upon updates or Announcements
5. Admin will be capabable to publish Announcements.
6. Students emailed the Announcements and the Dashboard Display
7. Student will be filling their info on CV form then the system generates CV
8. System will be embed with a Chatbot for FAQ Assistance
9. Administrators can search and retrieve archived documents efficiently.

### **4.2.2 Non-functional Requirements**

1. **Usability:** The system interface must be intuitive and easy to navigate for all users.
2. **Reliability:** The system must maintain an uptime of **99.9%**, with minimal downtime for maintenance.
3. **Security:** All user data must be encrypted and protected against unauthorized access. Multi-factor authentication should be implemented for sensitive actions.
4. **Scalability:** The system must support increasing numbers of users and document requests without performance degradation.

### **4.2.3 Performance Requirements**

1. **System Responsiveness: :** All key actions (Letter Request and Generation of CV ) must be completed in less than **5 seconds**.
2. **Concurrent Users :** The system must support at least **500 simultaneous users** without slowing down.
3. **Data Storage Capacity:** The database must accommodate a minimum of **1TB of documents** securely.

### **4.2.4 Software and Hardware Requirements**

* **Hardware Requirements:**
* A hosting PC with a minimun of **8 GB RAM**, 256 **SSD storage**, and cloud-based backup.
* Client devices (PCs, tablets, smartphones) with internet access.

**Software Requirements:**

* **Operating System:** Ubuntu 22.04 or later.
* **Database:** Postgres for secure document storage.
* **Frontend Framework:** Angular and Flutter for a responsive user interface.
* **Backend Framework:** Spring Boot for efficient backend processing.
* **AI Chatbot:** chatbase based chatbot for instant student support.

### **4.2.5 Preliminary Product Description**

The new system will be a **web and Moile based platform** that centralizes document request handling, approval workflows, credential verification, and AI-powered assistance. It will ensure:

1. **Automation** of document submission, approval, and retrieval.
2. **Enhanced security** through Encryption, JWT and Access control.
3. **Instant notifications** to keep students informed about request statuses.
4. **AI based Chatbot Assistance** Reduces Administrative Workload

**4.3. System Modeling**System modeling is a crucial phase in software development that involves creating abstract representations of the system to be developed. For the UNIDOCS system, comprehensive modeling was conducted using Object-Oriented Analysis approach to ensure clear understanding of system requirements, functionality, and structure.

**4.4. Dataflow Diagram**

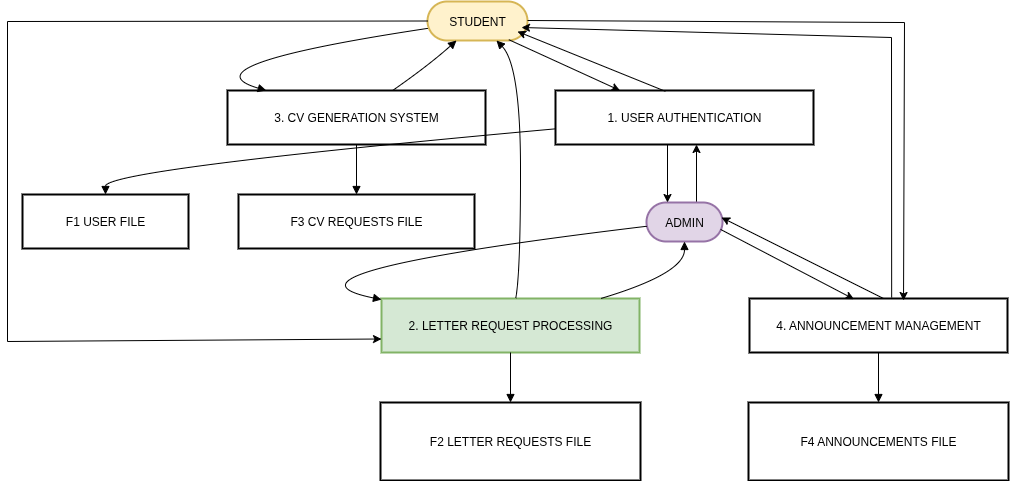
DFD describes the actual process that exists. The DFD that models the new system that you are going to develop may be different than this, and that shall come in the Design phase in Chapter 5. The DFD should be followed by a data dictionary, that unambiguously describes the format of each and every piece of information both in transit as well as in repository.

Figure 5: Dataflow Diagram

## **4.5 Requirement Structuring**

For the UNIDOCS system, this phase involves transforming the identified functional and non-functional requirements into detailed object-oriented models that capture the system's behavior, structure, and interactions.

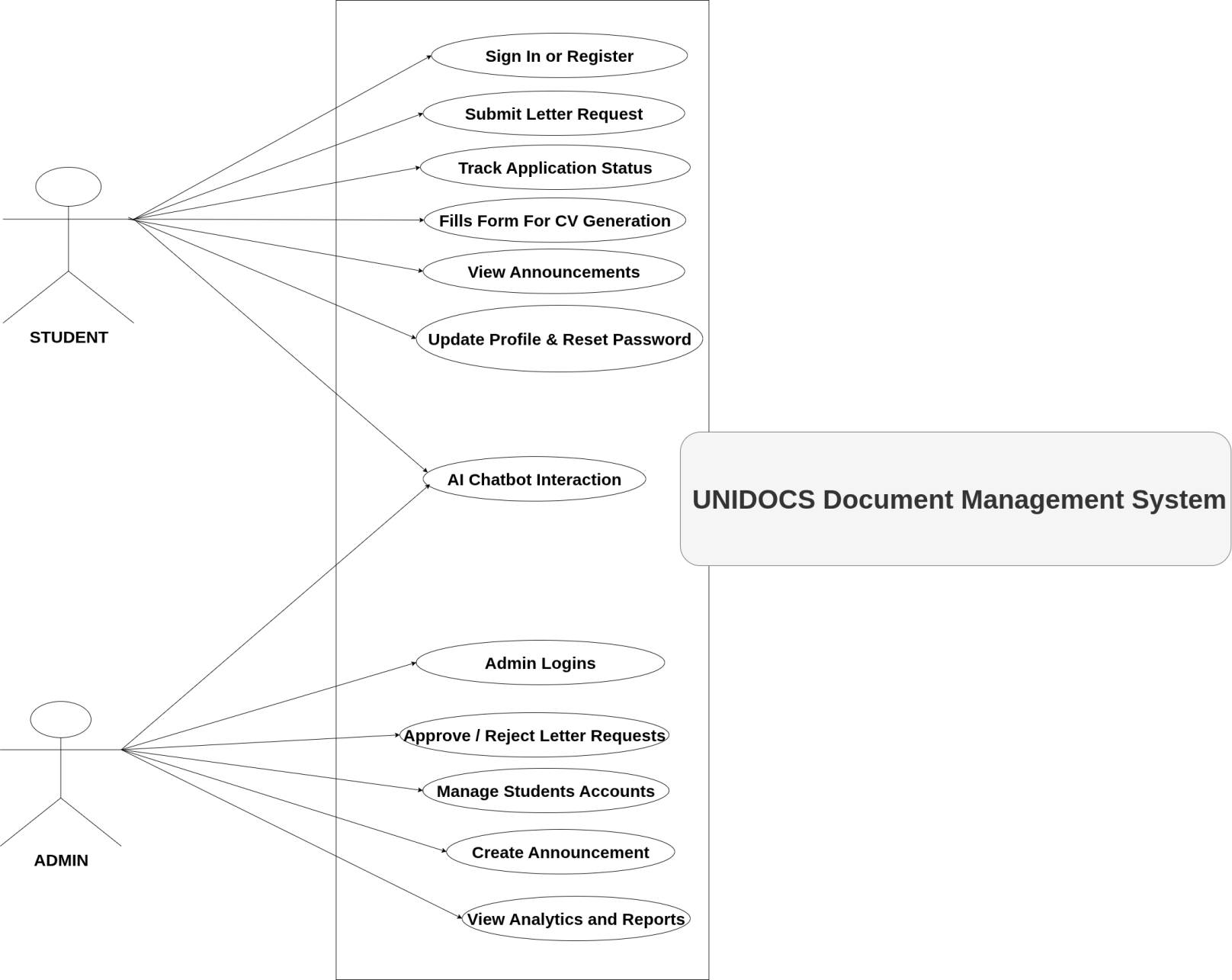
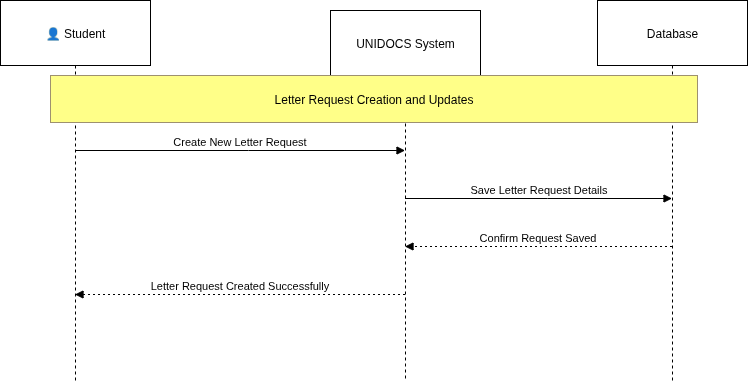
**4.5.1 Use case diagram**  
  


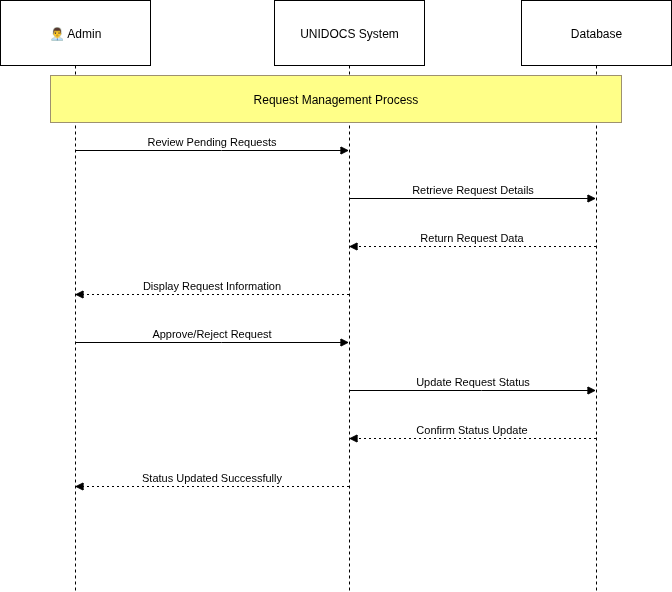
Figure 6: Use case diagram

**4.5.2. Use Case Documentation**  
  
Student Use Cases:

Students can register for an account, log in, and manage their profiles. They are able to submit letter requests by filling out a form, after which the system validates and stores the request, notifying the student of successful submission. Students can track the status of their applications, generate CVs using System templates, and download the resulting documents. They also have access to interact with the **Chatbot** , view announcements published by admins and can reset their passwords if needed. For each action, the system provides clear feedback, and in cases of errors (such as invalid input or missing files), appropriate messages are displayed to guide the student.

Admin Use Cases:

Admins log in to access the administrative dashboard, where they can manage student accounts and review all incoming letter requests. They are responsible for approving or rejecting these requests, with the system updating statuses and notifying students accordingly. Admins can create and publish announcements, ensuring important information reaches all users. They also have the ability to generate system reports. Throughout these processes, the system ensures that only authorized admins can perform sensitive actions, and provides clear notifications and error handling to support efficient management.  
  
**4.5.3 Sequence Diagram**



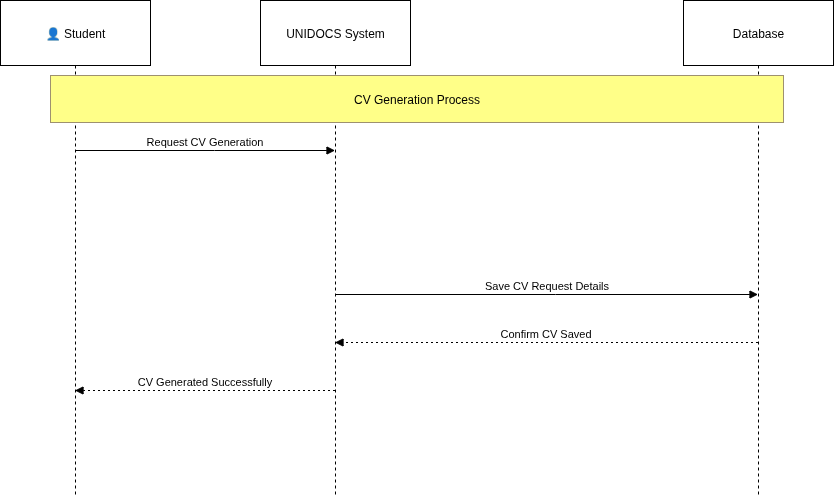
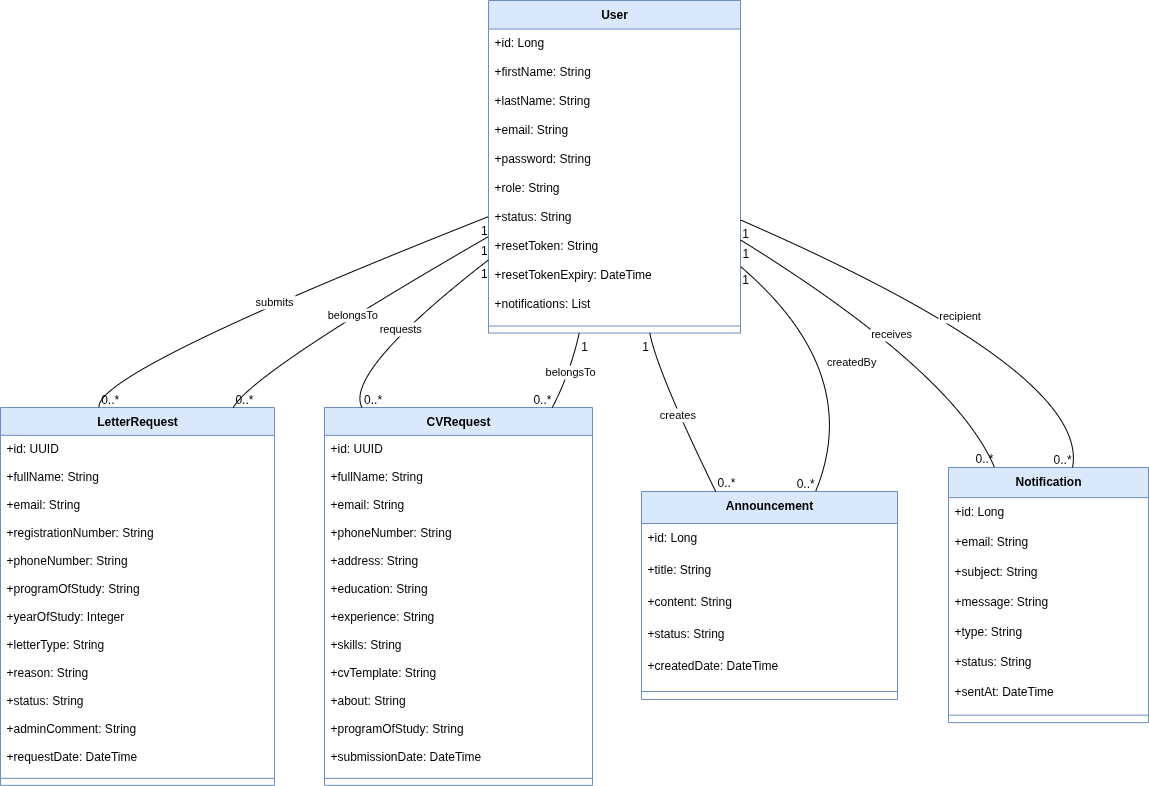


Figure 7:Sequence Diagram

******4.5.4. Conceptual modelling: Class diagram**

*Figure 8**: Class diagram*

### **4.5.5. Entity Relationship diagram**

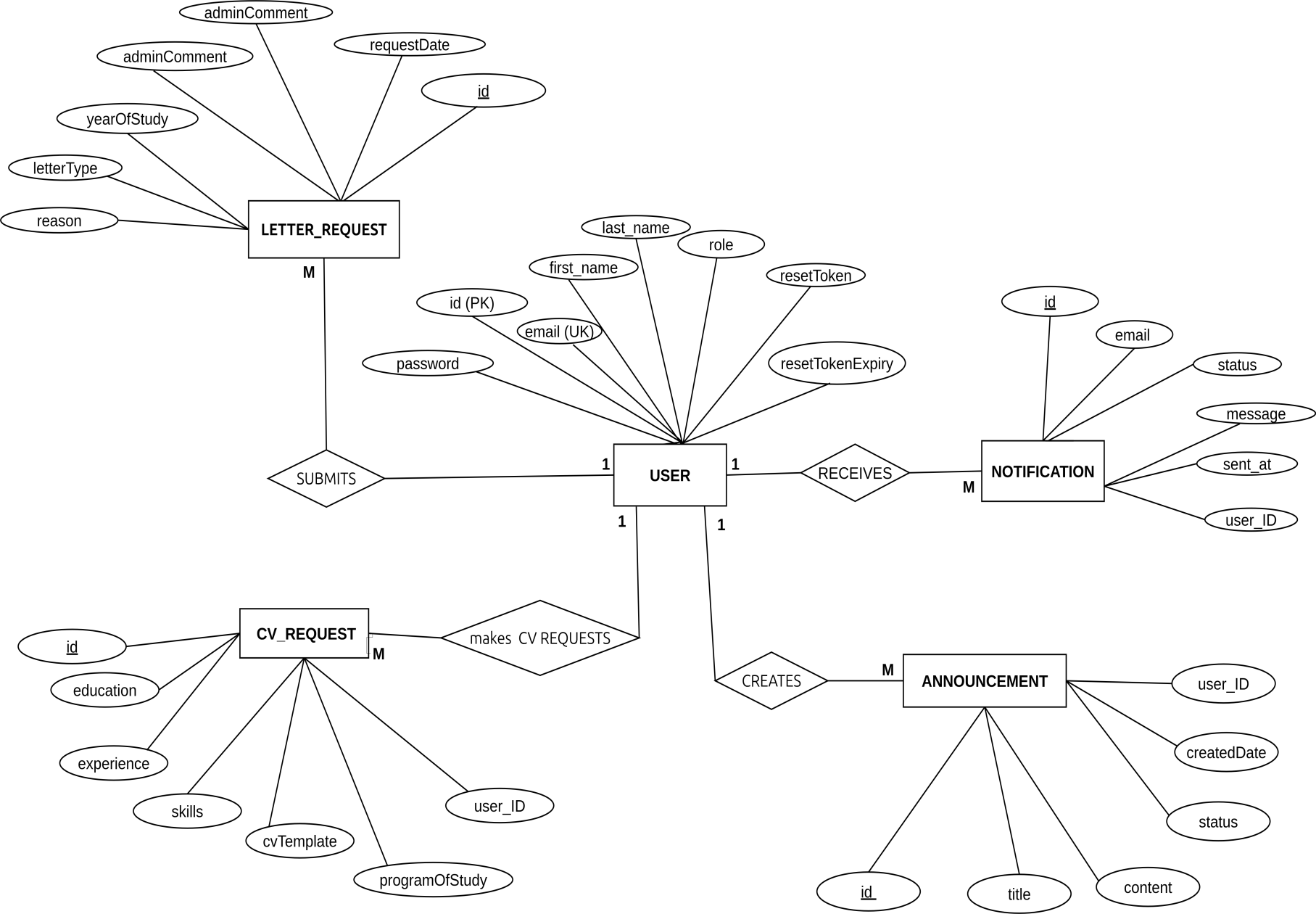


Figure 9: Entity Relationship diagram

**CHAPTER 5:**

# **References**

* Smith, J., & Brown, T. (2022). "Digital Transformation in University Administration." *Journal of Higher Education Systems*, 35(4), 145-160.
* Patel, M., & Jones, L. (2021). "Automated Document Management: Best Practices for Universities." *International Journal of AI and Data Science*, 18(3), 88-105.
* Kumar, R. (2023). "Enhancing Efficiency in Academic Document Processing." *Tech and Education Journal*, 27(2), 55-75.
* Postgres. (2022). "Postgress Documentation." Retrieved from https://postgres.com/doc/
* TensorFlow. (2023). "AI Chatbot Implementation Guide." Retrieved from <https://www.tensorflow.org/tutorials>
* Smith, J., Brown, T., & Lee, C. (2021). "Digital Document Management in Higher Education." *Journal of Information Systems*, 45(3), 112-126.
* Jones, R., & Patel, M. (2020). "AI Chatbots in University Administration: Enhancing Efficiency and Accessibility." *International Journal of AI Research*, 12(4), 88-102.
* Ellucian. (2022). "Banner Overview." Retrieved from https://www.ellucian.com
* M-Files. (2022). "Metadata-Driven Document Management." Retrieved from https://www.m-files.com
* EduSec. (2022). "EduSec Student Information System." Retrieved from <https://www.edusec.org>
* Smith, J., Brown, T., & Lee, C. (2021). "Digital Document Management in Higher Education." *Journal of Information Systems*, 45(3), 112-126.
* Jones, R., & Patel, M. (2020). "AI Chatbots in University Administration: Enhancing Efficiency and Accessibility." *International Journal of AI Research*, 12(4), 88-102.
* Ellucian. (2022). "Banner Overview." Retrieved from <https://www.ellucian.com>
* M-Files. (2022). "Metadata-Driven Document Management." Retrieved from <https://www.m-files.com>
* Brown, T., & Lee, C. (2022). "AI-Powered Chatbots in University Administration." Journal of Information Systems, 48(2), 200-215.
* Jones, R., & Patel, M. (2021). "Improving University Document Management Through Automation." International Journal of AI Research, 15(1), 102-118.
* AWS. (2023). "Cloud Computing and Scalability Solutions." Retrieved from [https://aws.amazon.com](https://aws.amazon.com/)
* GitHub. (2023). "Collaborative Software Development." Retrieved from [https://github.com](https://github.com/)