WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

PROJECT TITLE: AUTOMATED CLEARANCE MANAGEMENT SYSTEM FOR WOLKITE UNIVERSITY

NAME	ID NO
EYADER TSEHAYU	NSR/0559/12
EYOB KEFALE	NSR/0569/12

ADVISOR: MR. BEREKET SIMON (MSc.)

Date - 10/05/2024.

Wolkite University, Wolkite, Ethiopia

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

PROJECT TITLE: AUTOMATED CLEARANCE MANAGEMENT SYSTEM FOR WOLKITE UNIVERSITY

SUBMITTED TO DEPARTMENT OF SOFTWARE ENGINEERING IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

NAME	ID NO
EYADER TSEHAYU	NSR/0559/12
EYOB KEFALE	NSR/0569/12

ADVISOR: MR. BEREKET SIMON (MSc.)

Date - 10/05/2024.

Wolkite University, Wolkite, Ethiopia

Declaration

This is to declare that this project work which is done under the supervision of Mr. Bereket Simon, and having the title “Automated Clearance Management System for Wolkite University” is the sole contribution of:

1. Eyader Tsehayu
2. Eyob Kefale

No portion of the project work has been illicitly duplicated, constituting plagiarism. All referenced sections have been employed to support the ideas and have been appropriately cited. We acknowledge responsibility and accountability for any repercussions should a breach of this declaration be substantiated.

Date: _____

Group Members:

Full Name	Signature
Eyader Tsehayu	_____
Eyob Kefale	_____

Approval form

This confirms the receipt of the project software requirement specification document titled “Automated Clearance Management System for Wolkite University” from Eyader Tsehayu and Eyob Kefale by the Department of Software Engineering at Wolkite University's College of Computing and Informatics. Both the advisor and the Department of Software Engineering have endorsed this senior group project.

Advisor and Department Head Approval Form

Advisor Name	Signature	Date
_____	_____	_____
Department Head Name	Signature	Date
_____	_____	_____

Examiners Approval Form

Examiner Name	Signature	Date
_____	_____	_____
_____	_____	_____
_____	_____	_____

Acknowledgment

First and foremost, we express our sincere gratitude to GOD for His constant presence in every aspect of our lives. His boundless love and unwavering mercy have been steadfast companions, guiding us through both the highs and lows, even in moments when we feel undeserving of such blessings.

A big thanks goes to Wolkite University's College of Computing and Informatics for providing us this opportunity to embark on our final project. This chance has significantly enhanced our learning experience, and we are truly grateful for it.

We also want to acknowledge and appreciate Mr. Bereket Simon, our advisor, for his insightful intellectual guidance on our task responsibilities. His valuable advice and consistent support have been indispensable throughout our journey.

In closing, we extend our deep appreciation to our families for their unwavering support. Their encouragement has played a pivotal role in helping us successfully fulfil the mission set forth by Wolkite University.

Table of Contents

List of Tables

List of Figures

Abbreviations

WKU - Wolkite university

WKUCMS – Wolkite university clearance management system

CMS – Clearance management system

Chapter 1

1.Introduction

1.1. Introduction

Clearance is a status granted to individuals, typically members of the military, university students and employees of governmental and non-governmental organizations and their contractors, if they fulfill certain rules and regulations regardless of the services they get from their organization and allowing them to access their organization services.

Clearance, in the context of a Clearance Management System, refers to the formal process of obtaining approval or authorization for a particular action or access within an organization. It involves the systematic verification and validation of various criteria to ensure that an individual or entity meets the specified requirements or conditions necessary to proceed with a specific request. Clearances are crucial in maintaining security, compliance, and order within diverse institutions, including governmental organizations, universities, hospitals, offices, and non-governmental organizations. Clearance Management System is a system which enables the organization to generate and manage a clearance approval form for their clients. It is used in different governmental organizations such as universities, hospitals, offices and in non-governmental organizations.

Wolkite University, as a higher education institution, encompasses a multitude of both manual and automated processes. Presently, the university employs a manual Clearance System designed to facilitate the clearance of individuals regarding their material and infrastructure usage. The existing manual clearance process necessitates requesters to physically traverse the university, securing signatures on clearance forms from various officers, and repeatedly queuing to submit or obtain additional clearance approvals. Recognizing the inefficiencies inherent in this manual process, our project focuses on automating the Clearance Management System at Wolkite University. The automated system aims to address the challenges associated with manual processing by minimizing manpower usage, reducing errors, saving time, and enhancing the overall clearance approval experience for requesters.

This project follows a structured approach. Initially, we will identify and define the key stakeholders of the system, eliciting their requirements. Subsequently, we will classify, categorize, specify, and validate the gathered requirements. Ultimately, the documented requirements will serve as a foundation for the development of the Automated Clearance Management System.

1.2. Statement of Problem

The university currently faces numerous challenges due to its reliance on manual methods for managing and performing both students and staffs clearances. The clearance process is characterized by its time-consuming and arduous nature, taking over three days to complete and causing significant stress for both staff and requesters. This delay in processing stands out as the primary issue. Additionally, High file redundancy throughout the clearing process results in information being duplicated across multiple sites without centralized control.

This lack of a unified information source results in inefficiencies in data management. The necessity for every university member to physically visit each office consecutively with a clearance form presents challenges, especially when key staff members are unavailable. Manual documentation of clearance forms further increases the risk of vital document loss, relying on cumbersome manual processes for filing. The existing manual clearance system demands a substantial workforce, contributing to high operational costs. These challenges have persisted for a decade at Wolkite University, prompting the project team to design a system aimed at addressing these issues comprehensively. By reducing manpower, time, and space requirements, the project endeavors to alleviate the longstanding challenges associated with the manual clearance system at WKU.

1.3. Objective

1.3.1. General Objective

The general objective of the project is to develop an automated clearance management system for Wolkite University.

1.3.2. Specific Objective

- To specify user's requirement for potential problems identification.
- To generate alternative solution for identified problems.

- To encode selected proposed solution.
- To develop student's, staff's and administers data definition and manipulation module.
- To develop user management module.
- To develop account and role management system.
- To develop report generation and certification system.

1.4. Sustainability of project

1.4.1. Feasibility testing

Technical Feasibility

The proposed automated clearance management system, will be developed using Next.js and related libraries, underscores a robust foundation for streamlined development. The project requires a proficient team of technical experts adept in modern system development techniques to leverage the capabilities of Next.js, ensuring efficient server-side rendering, optimal performance, and compatibility with existing university infrastructure. With a focus on scalability, seamless integration, and the implementation of robust security measures, the project aligns with contemporary development practices. The user interface design prioritizes responsiveness and intuitiveness, providing stakeholders with an enhanced and user-friendly experience. Continuous learning and adaptation to emerging technologies further characterize the project's technical approach, ensuring its viability in addressing the complexities of the Wolkite University clearance system.

Operational Feasibility

The system to be developed provide accurate, active, secured service and decreases labor workers and the new system which supports the major activity of the circulation diction will take advantage to solve the problem outlined in the statement of the problem section. Moreover, the user can operate the system with little training. Thus, it can be said that it is operationally feasible.

Economic Feasibility

The project is economically feasible because developing such systems by its own students saves much more cost than giving it to other developers or buying it. Tools that the proposed system

requires are almost available in the university, so the organization is profitable by using this system.

1.5. Scope and limitation of project

1.5.1. scope of the project

The scope of this project covers the process related to student, staff and admin clearance in Wolkite university. This system performs tasks like the

- **Workflow Automation:** Automated workflows for clearance processing, reducing manual interventions.
- **Online request and approval:** Users can submit clearance requests, eliminating the need for physical paperwork.
- **Real-time Tracking:** Users can monitor the status of their clearance requests in real-time, fostering transparency.
- **Notifications:** Automated notifications to keep users informed about the progress and completion of their clearance requests.
- **Document Management:** Digitized storage of clearance-related documents for easy retrieval and auditing.
- **Reporting and Analytics:** Comprehensive reporting tools for administrators to analyze clearance data and identify trends.

1.5.2. Limitation of project

While this project endeavors to streamline and automate clearance processes, it is essential to acknowledge certain limitations. Notably, the system does not encompass the registration of records and associated properties. Consequently, the verification of records will necessitate a manual process, distinct from the automated workflow implemented by the project. This limitation implies that certain aspects of the clearance validation, specifically those related to record checking, will require manual intervention outside the system's scope. It is crucial for users and stakeholders to be aware of this constraint when engaging with the Automated Clearance Management System.

1.6. Significance of the Project

The significance of this project lies in its transformative impact on the Wolkite University clearance system, bringing about substantial improvements and addressing longstanding challenges. By transitioning from a manual to a computerized system, the project aims to streamline and enhance the entire clearance process. The key significance includes:

- **Efficiency Enhancement:** The implementation of the automated clearance system will significantly reduce processing times, providing a more expedited and responsive experience for users. This efficiency improvement is crucial in minimizing delays and enhancing overall operational effectiveness.
- **Workload Reduction:** The project endeavors to alleviate the workload on stakeholders involved in the clearance process, including students, staff, and administrators. Automation will replace labor-intensive manual tasks, allowing stakeholders to focus on more value-added responsibilities.
- **Real-Time Information Access:** Moving from a manual to a computerized system ensures that users have access to real-time information regarding their clearance status. This transparency fosters a more informed and empowered user experience, reducing uncertainty and improving communication.
- **Error Reduction:** The automated system is designed to mitigate errors associated with manual processing. By implementing systematic checks and validation processes, the project aims to enhance accuracy and reduce the likelihood of mistakes in clearance procedures.
- **Resource Optimization:** The project's significance extends to resource optimization by reducing the reliance on manual efforts, minimizing paperwork, and contributing to a more sustainable and environmentally friendly approach to clearance management.
- **Enhanced User Experience:** The introduction of a computer web application promises a user-friendly interface, making clearance processes more accessible and intuitive. This enhanced user experience is instrumental in fostering positive interactions between the system and its users.

- **Strategic Decision-Making:** With improved data management and reporting capabilities, administrators can make more informed and strategic decisions. The system's ability to generate insightful reports and analytics contributes to better overall governance and management of clearance-related activities.

The significance of this project lies in its ability to revolutionize the clearance management system at Wolkite University, ushering in a new era of efficiency, transparency, and user satisfaction.

1.7. Beneficiaries of the project

This project has many beneficiaries to users of the system. The following are some of the beneficiaries of the project:

1. **Students:**
   - Streamlined Processes: Students will experience streamlined and expedited clearance processes, reducing the time and effort required for administrative procedures.
   - User-Friendly Access: The automated system ensures a user-friendly interface for students to submit clearance requests, monitor progress, and receive timely updates.
2. **Staff Members (Including Admins):**
   - Workload Reduction: Both regular staff members and administrators will benefit from reduced manual tasks, allowing them to allocate time more efficiently and focus on core responsibilities.
   - Administrative Empowerment: Administrators, as a subset of staff, will have enhanced tools for managing and overseeing clearance processes, contributing to improved administrative efficiency.
3. **Staffs with Approval Privilege:**
   - Efficient Approval Processes: Staff members with approval privileges will experience a more efficient clearance approval process, enabling them to review and approve requests with ease.

- Access to Comprehensive Information*:* Approval staff will have access to comprehensive information about clearance requests, facilitating informed decision-making.

4. **University as a Whole:**
    - Operational Efficiency: The project contributes to the overall operational efficiency of the university by introducing a modernized and automated clearance management system.
    - Cost Savings: Through reduced manual efforts and optimized resource allocation, the university can achieve cost savings and improve resource utilization.
    - Enhanced Image: A streamlined and technologically advanced clearance system enhances the university's image, showcasing a commitment to efficiency and innovation.

The project's beneficiaries encompass a wide spectrum of the university community, aiming to enhance the experiences and operational effectiveness for students, staff members (including administrators), staffs with approval privileges, and the university.

1.8. Methodology

1.8.1. Data collection

In developing a clearance system for Wolkite University, the following development methodologies and tools will be applied. To gather accurate data from the concerned body the project team will made excessive effort by interviewing, Observing and document revising, but the deepest takes brainstorming (since it helps to make the part of the Solution).

1.8.2. Analysis and design

In this project the team will use Object Oriented System Analysis and Development methodology (OOSAD). This has two phases. Object Oriented Analysis (OOA): - During this phase the team used to model the functions of the system (use case modelling), find, and identify the business objects, organize the objects and identify the relationship between them and finally model the behavior of the object. Object Oriented Design (OOD): - During this phase the team

used to refine the use case model to reflect the implementation environment, model object interactions and behaviors that support the use case scenario, and finally update object model.

1.8.3. Tool and Methodology

Frontend Technologies

For the development of the user interface, we will leverage the capabilities of HTML, CSS, SCSS, and JavaScript, with a primary focus on harnessing the power of Next.js. As a full stack React framework, Next.js offers enhanced scalability and maintainability, providing an optimal solution for crafting a dynamic and responsive frontend.

Backend Technologies

In the backend infrastructure, our technology stack will include Next.js, a powerful full-stack framework built on top of React and Node JS with Express JS framework, enabling seamless integration between frontend and backend components. Additionally, MongoDB will serve as our database solution, aligning with the business logic requirements and ensuring efficient data management within the system.

1.8.4. System Testing Methodology

The development team will perform comprehensive testing encompassing both functional and non-functional aspects prior to the deployment of the automated clearance system. Functional testing will verify the fulfilment of all requirements, while non-functional testing will concentrate on security, reliability, and system robustness. The objective is to release flawless software that aligns with user expectations, functions seamlessly, and upholds elevated standards of usability and security.

Unit Testing

Unit testing involves testing individual components or functions of the code in isolation to ensure they work as intended. For our project, we will use testing frameworks Jest, which is commonly used for JavaScript and React applications. Jest provides a simple and effective way to write and run unit tests for React.js components and functions.

Integration Testing

Integration testing focuses on testing the interactions between different components or modules to ensure they work together correctly. In our project, we will use Jest to perform integration tests. Jest supports testing asynchronous code, making it suitable for testing the integration of different parts of your application.

System Testing

System testing involves testing the entire system as a whole to ensure that all components and modules work together seamlessly. In our project, we will use end-to-end testing framework Cypress to simulate user interactions and test the complete application flow.

Chapter 2

2. Description of existing system

2.1. Introduction to existing system

Wolkite University currently employs a manual clearance system for managing the clearance of students and staffs. The process involves individuals obtaining clearance forms from various university stakeholders, including the registrar and the human resource office. Students acquire clearance forms when submitting withdrawal forms, at the end of each academic year, or upon graduation. Similarly, staff members obtain clearance forms when leaving the university or receiving scholarships. This manual system is intricate and time-consuming, requiring users to visit different clearance offices to collect signatures, indicating that they have returned borrowed materials and have no outstanding debts.

Before signing the clearance form, certain officers, such as technical assistants, dormitory chiefs, sports and recreation heads, bookstore keepers, cafeteria chiefs, and those in finance and academic affairs, review the borrowed properties or agreements directly associated with the university. If all borrowed items are returned, they proceed to sign the clearance form; otherwise, they abstain from signing. Other officers, including college deans, department heads, library chiefs, student deans, legal service directorate directors, and human resource personnel, sign the clearance form by assessing the hierarchical structure, as they lack direct involvement with university properties.

The process involves utilizing paper documents containing borrower information, such as names, identification numbers, and categories of borrowed materials, to maintain a record of students who have borrowed items. These clearance forms are stored in file cabinets, resulting in inefficiencies in data management, including challenges in processing and analyzing data. Retrieving specific student clearance forms requires manual searches in the file cabinets, contributing to data management issues within the current system.

The clearance process at Wolkite University is intricate and methodical, requiring students to navigate through several meticulous steps for a successful campus clearance. The initial step involves obtaining a clearance form from the registrar's office, setting the foundation for the

subsequent stages. Following this, the department head meticulously reviews and approves the clearance, ensuring that all necessary requirements are met by the student. The academic dean further validates the process by verifying the department head's signature, enhancing the thoroughness of the clearance. The student then seeks approval from the cafeteria chief, who, in turn, scrutinizes the presence of an ID card and endorses the form upon fulfillment of the criteria.

The library becomes the next checkpoint, where the college bookstore keeper confirms the return of textbooks and related materials before granting clearance. If approved, the library chief of circulation acknowledges the student's request and signs the form. The student proceeds to the sport and recreation office, where the head of the office assesses the return of sporting goods before endorsing the clearance. Subsequently, the dormitory chief examines the return of borrowed dormitory items, including keys, furniture, and other essentials, ensuring the completeness of the clearance process. Finally, the dean of students consolidates the approvals from the dormitory, cafeteria, and sport and recreation offices, addressing discipline-related concerns, and endorses the clearance.

Upon completion of this comprehensive process, the student submits one form to the registrar's office, another to the department head, and retains the remaining for personal records, marking the successful conclusion of the campus clearance. However, the complexity and lengthiness of this procedure often result in delays for both students and officers. Some steps may be skipped or not followed correctly due to the cumbersome nature of the process, highlighting the need for a more streamlined and efficient clearance system.

Similarly, staff members face a parallel labyrinth of steps to complete their campus clearances. The detailed scrutiny by various offices demands significant time and effort, contributing to the overarching challenges faced by both students and staff in navigating the bureaucratic intricacies of the clearance process at Wolkite University.

2.2. Drawback of the existing system

The major drawback of the existing manual systems is:

- **Energy and time wastage**: Existing manual clearance systems consume considerable time and effort from both students, staff, and officers.

- **Unnecessary office visits:** Students and staff are compelled to visit specific offices even if they lack direct affiliations with the resources in question.
- **Unnecessary resource usage:** Traditional methods contribute to avoidable waste, such as paper and pen consumption, highlighting inefficiencies.
- **Lack of organized data:** Offices often lack well-organized data, leading to haphazard clearance approvals without thorough verification of student status.

2.2.1. Performance (Response time)

- **Wait in the Queue:** The manual clearance system at Wolkite University results in prolonged wait times for individuals processing clearance forms, standing in queues for an extended duration.
- **Staff Unavailability:** The unavailability of key staff during the clearance form processing adds to delays and inefficiencies in the overall clearance procedure.
- **Time-Consuming Retrieval:** Retrieving specific clearances from respective offices is a time-consuming process, contributing to overall delays.

2.2.2. Information Input and Output

In the manual system of Wolkite University's clearance system:

- **Human-Recorded Data:** Data records are manually entered by human operators, introducing the potential for errors and inaccuracies.
- **Inaccurate Information:** The manual input of information regarding students and staff members may lead to inaccuracies and discrepancies.
- **Redundancy and Inflexibility:** The manual inputs are often redundant and inflexible, resulting in an output that may not accurately reflect the current state of affairs.

2.2.3. Security and Control

The existing system lacks much security and controls, as indicated below:

- **Exposure to Theft:** Manual files are susceptible to theft, with the risk of students signing forms in place of authorized officers, potentially leading to data loss.
- **Insecurity of Student Data:** The security of student data is compromised within the manual system, posing risks of unauthorized access and potential breaches.

- **Unrestricted Access:** Accessibility to student data on manual forms is not controlled, allowing for unrestricted access by individuals.

2.2.4. Efficiency

The existing system has low efficiency in speed, time conception, and resource utilization due to its manual nature. The inefficiencies include*:*

- **Time-Consuming Tasks:** Manual processes result in time-consuming task execution, contributing to delays in the overall clearance workflow.
- **Redundant Data Recording:** Redundancies in data recording can occur, leading to inefficiencies and potential inaccuracies.
- **Error-Prone Data Entry:** Manual data entry procedures are prone to errors, impacting the accuracy and reliability of the clearance system.
- **Inflexibility:** The system's inflexibility hinders the ability to adapt or modify data efficiently when needed, leading to further operational challenges.

2.3. Users of the Existing System

The intricacies of Wolkite University's current clearance system are navigated by a diverse set of users, each contributing uniquely to the clearance process. These users, or stakeholders, are classified into two distinct categories: those associated with student clearance and those linked with staff clearance. The key stakeholders include:

Users related to student clearance:

- **Students:** These are the learners within the university who initiate the clearance process. Students receive the clearance form from the registrar's office, triggering the entire clearance workflow.
- **Registrar:** The office responsible for providing clearance forms to students and receiving the completed clearance forms at the conclusion of the clearance process.
- **Head of Department:** This individual ensures that students have returned all educational-related equipment, including computers, electronics, laboratory equipment, and uniforms, among others.

- **Academic Dean:** The academic dean of the college approves the signing of the department head, adding a layer of verification to the clearance process.
- **Cafeteria Chief:** Head of the cafeteria who verifies and approves a student's clearance, ensuring the blocking of cafeteria services until the clearance is completed.
- **College Bookstore Keeper:** A stakeholder who checks for the return of textbooks and guides before granting clearance.
- **Library Chief of Circulation:** The person in charge of library circulation who approves the signing of the College Bookstore Keeper in the clearance process.
- **Sport Head:** Head of the sport and recreation office who ensures the return of all sports materials such as kits and balls.
- **Dormitory Chief:** Oversees dormitory services and checks the return of dormitory materials, including keys, furniture, bulbs, mattresses, and other essentials.
- **Dean of Students:** Approves the clearance process for three offices (dormitory, cafeteria, sport, and recreation) and addresses discipline issues related to the student.
    - Except for students, all others involved are employees working within the university and serve as stakeholders in the clearance process.

Users related with staff and admin clearance:

- **Library director**: - approves the staff clearance that is signed by the library circulation experts.
- **Accountant**: - the person who assures the staff is free from any payments like allowance, suspense and advance payments that are given for different reasons before the work is completed.
- **Main treasurer**: - is who approves the clearance by checking the signing of the accountants.
- **WKU money and saving officer** checks the staffs profile that is related with saving and borrowing of money from the association and approves the clearance if he/she is free from any credit.

- **Finance directorate director**: other stakeholder that checks the approval of clearance from offices that is related with finance. Like accountants, main treasurer and WKU money and saving officer then the director will approve the clearance directly.
- **Fixed items storekeeper**: person who monitors all items that are given staff members of the university. There are three different storekeepers.
    - The first one is related with furniture materials,
    - The second with computers and other electronics equipment,
    - The last storekeeper is doing with ender or semi ender items.

  All this three will check the staff history of borrowing the above materials and they approves the clearance.
- **Buying and property management directorate director**: will approve the clearance by checking the signing of the fixed item storekeepers.
- **WKU teachers' association**: the head of the association who confirms the teacher has not any issue or debt related with the association.
- **Accountant of the Teacher and staff saving association**: - The person who checks for returned debits before approving the clearance.
- **Chairmen of the Teacher and staff saving association: -** one of the stakeholders who manage the association checks preceding office approval and approves teachers' clearance request.
- **Registrar secretary office: -** The current system stakeholder that approves and rejects clearance request of the teacher and staff by checking the approval of the preceding office.
- **Student Dean secretary office: -** one of the stakeholders of the current system approve and reject staff and teacher clearance request based on the approval of the preceding office.
- **Facility development directorate: -** approve and reject approval request of the teacher and staff by checking that they are free from any facility related issues.
- **Community service directorate: -** one of the universities stakeholders approve and reject approval request of the teacher and staff by checking that they are free from any research or community service related things. And they check the sequence requests.

- **Research directorate director: -** One of the current system stakeholders that approve and reject the clearance request of each staff and teacher by checking the approval of the preceding office and, they should be free from any research related works.
- **Research and community service vice president: -** Approve and reject clearance request of the staff and teacher based on the approval of the preceding offices.
- **ICT directorate -** One of the stakeholders of the current system that approves and reject the clearance request of the staff and teacher based the approval the preceding office and deactivate their account which have the universities domain.
- **Academic affair vice president** - Before approving the clearance, the person checks academic agreements with teachers and surety for other teachers.
- **Internal audit**: - The person who checks internal properties and property officer approvals.
- **Legal service directorate director** - The person who checks the Academic Affairs Vice President's approval before approving the clearance.
- **Ethics and anticorruption** - The person who investigates applicants' ethical issues.
- **Administration corporate management vice president** - The person who checks the applicant's identification card with his/her full name.
- **Record officer** - Using the applicant's profile, the person who checks if the staff is free from any warrant or debt issues.
- **Human resources directorate director**- The person who initiates the clearance process for staff applicants and approves them based on the hierarchy.
    - The users related with staff and admin clearance process at Wolkite University involves a complex network of 29 stakeholders, each playing their own role in ensuring a thorough and accurate clearance procedure. But the number of stakeholders which are listed above is less than 29 this occurs because of the position of the Library Chief of Circulation has been consolidated to eliminate redundancy and there are some offices which do same works in different positions or locations like the storekeeper and library Chief of Circulation officers.

2.4. Major functions/activities in the existing system

2.4.1. Input Analysis

In the current system, various forms serve as inputs, encompassing crucial information such as the student's full name, ID number, department, college, year, semester, and the reason for withdrawal. Students initiate the process by completing these forms, which are then submitted to the respective offices. To receive official receipts, students must meticulously provide these documents, ensuring accurate and comprehensive input.

2.4.2. Process Analysis

The workflow involves students completing clearance forms, a pivotal step that triggers subsequent actions. Once filled in, these forms are collected and sequentially signed by the appropriate offices. The signing process serves as tangible proof that the student has fulfilled all necessary requirements for clearance. Each signature signifies the validation of different aspects, leading to the issuance of a certificate. This certificate serves as conclusive evidence that the student has successfully completed the clearance processing, paving the way for their departure from the university.

2.4.3. Output Analysis

The culmination of the clearance process results in a tangible output – a certificate or an equivalent clearance document. This document is handed to the student or the staff, explicitly stating that they have fulfilled all university requirements. The certificate serves as official confirmation that the student is now free to leave the university, marking the successful conclusion of the clearance process. It not only holds significance for the student or the staff but also acts as an official record, streamlining administrative procedures related to the student's departure from Wolkite University.

2.5. Business rules of the existing system

Wolkite University operates under a set of meticulous business rules and regulations that dictate the processes and conduct of both students and officers during the clearance procedures. These rules serve as a comprehensive framework to ensure the proper and systematic clearing of customers, emphasizing adherence to established protocols and standards. Students engaging in the clearance process are obligated to follow specific guidelines, including accurately filling out required forms with essential details such as their full name, ID number, department, college,

year, semester, and reason for withdrawal. Moreover, the university imposes strict regulations on officers involved in the clearance process, mandating precise verification and documentation at each step of the clearance workflow. These business rules are designed to uphold the integrity of the clearance system, facilitate efficient processing, and ultimately contribute to a seamless experience for both students and staff at Wolkite University.

The main business rules or principles of the existing system are: -

BR1: - Anyone who signs clearance form must be the member of the university.

BR2: - Students should take a clearance form from their college registrar.

BR3: - Admin and staffs should take clearance form from the Human Resource directorate director office.

BR4: -Anyone who wants a clearance certificate should full fill his/her responsibility, requirement and should return any borrowed material or services before they request clearance approval.

BR5: - The officer of the university should check responsibilities, requirements, and any debt of anyone who request approvals.

BR6: - The officers of the university that are assigned to check clearances should approve or reject clearance requests anyone who requests according to their rule.

BR7: - In case of student clearance after the final approval from the Registrar, students are directed to retain one copy for their personal records, submit another copy to their respective department head, and ensure that the third copy is maintained within the Registrar's office. This procedural adherence guarantees that the student possesses documentary evidence of clearance for personal reference, while simultaneously furnishing the department and university with the necessary documentation for official records.

BR8: - In case of staff clearance after the final approval from the human resources directorate director, staff are directed to retain one copy for their personal records, submit another copy to the record officer, and ensure that the third copy is maintained within the human resources directorate director office. If the staff is admin staff his clearance started from his/her corresponding director and if the staff is academic staff the process starts from his/her

corresponding department head. This procedural adherence guarantees that the staff possesses documentary evidence of clearance.

2.6. Forms and other documents of the existing system

WOLKITE UNIVERSITY
OFFICE OF THE REGISTRAR
CLEARANCE/WITHDRAWAL FORM

Purpose:

Only with the proper termination below can transcripts. Letters of enrollment or honorable dismissal be issued. Readmission to the college will be considered if proper termination is certified by the registrar's office.

Procedure:

1. To be completed in triplicate form
2. Complete the first part of this form
3. Obtain the signatures designated in part II
4. Return this form to the registrar

Part I

1.1 Full Name _____ ID NO_____
1.2 Collage_____Department_____year_____semister_____
1.3 Last Date Class Attended_____
1.4 Reason for withdrawal_____(to be filled by registrar in case of withdrawal)

Part II	full name	signature
2.1 Department head advisor	_____	_____
2.2 Academic Dean (Include Date)	_____	_____
2.3 Cafeteria Chief	_____	_____
2.4 Library chief of circulation	_____	_____
2.5 Collage Book Store	_____	_____
(For return of text books and other equipment issued by the B.S)		
2.6 Sport	_____	_____
2.7 Dormitory Chief	_____	_____
2.8 Dean of students	_____	_____
2.9 Registrar	_____	_____

Date received_____/_____/_____

Figure 1. Form signed for student clearance process.

Wolkite University
መልቂጤ ዩንቨርሲቲ

ቁጥር ----------
ቀን ----------

Human resource M/D/Directorate
የሰዉ ሀብት አስተዳደር ልማት ዳይሬክቶሬት

የአካዳሚክ ሰራተኛ ንብረት ማስረከቢያ ቅጽ/ክሊራንስ/

ሠ/ የዚህ ንብረት ማስረከቢያ ቅጽ አገልግሎት አንድ የዩኒቨርሲቲው አባል፡

1. ስንብረት ቁጥጥር ሲባል በአዲያሰውንየዩኒቨርሲቲውንንብረትማስረክብቢናርበት
2. ከሌሎች ከንብረትና ገንዘብ አዳ ጋር በተገናኘ መረጃ ሲፈስግ
3. ስተመሰነ ጊዜስት/ት ወይም ስምርምር ከመደበኛ ስራው የመንግስት ንብረት በእጅ አስመናረን ማረጋገጫ
4. ሰራተኛው ስራውን በሚስቅበት ወቅት ክሲራንስ ከመሰጠቱ በፊት የመንግሰት ንብረትና ገንዘብ ያስረከቡ መሆኑን ማረጋገጫ

ስ/ የሰራተኛ ስም ከነአያት ----------

የስራ መደቡ መጠሪያ ----------የመ/ቁጥር ----------የስራ ደረጃ --------ደመወዝ ------

የስራክፍል ---------- የሰራተኛው ፊርማ ----------

ሐ/ የገደስ ንብረትና ገንዘብ ቢናር ተጠያቂ የሚሆን ዋስ ስም ከነአያት ----------

የሚሰራበት ስራ ክፍል ---------- የዋስ ፊርማ ----------

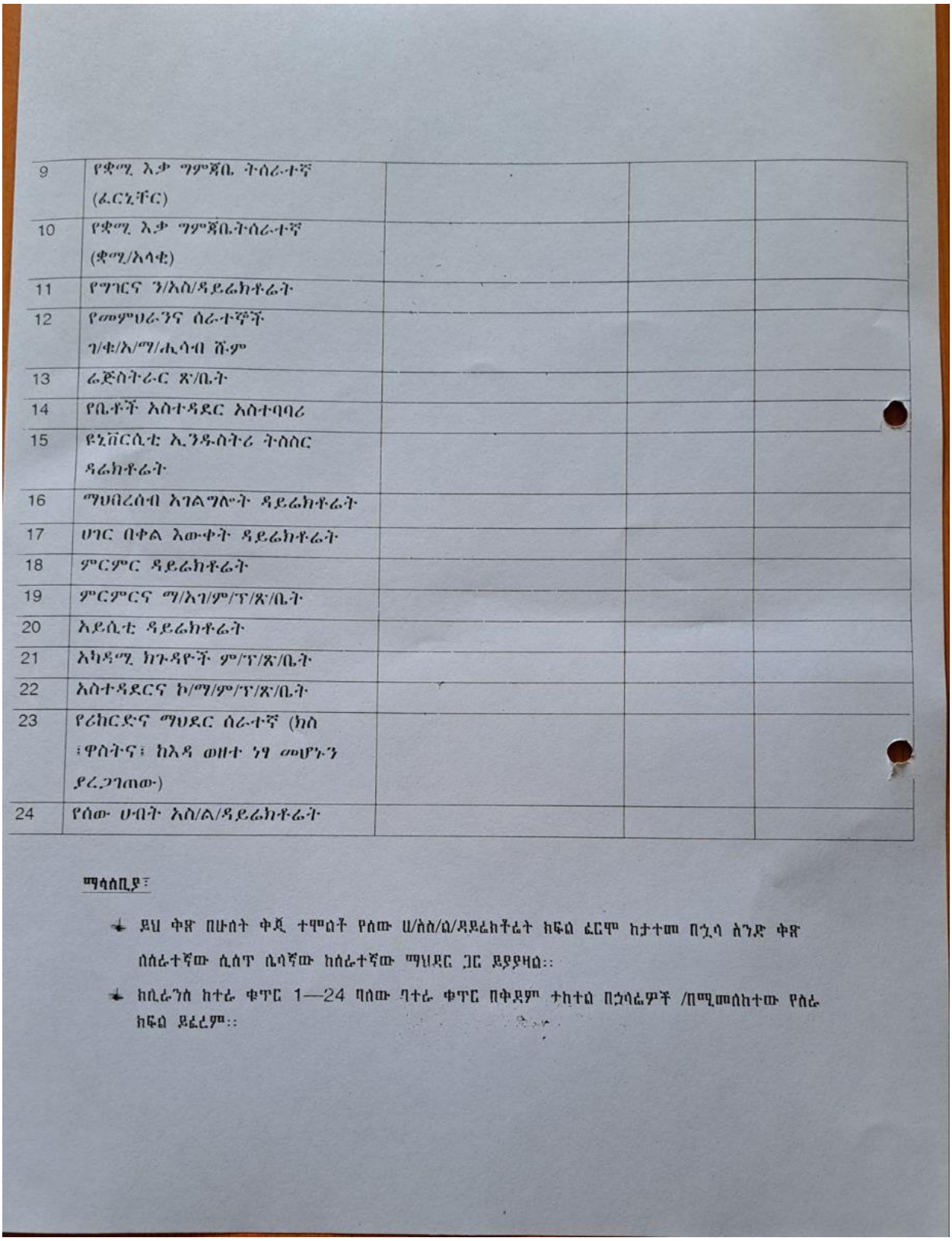
መ/ ከስራ ገበታ የሚስየበት ምክንያት----- ----------

ቀን ----------

ሠ/ቅጽ እንዲሞላ የፈቀደው ሓላፊ

- ስአካዳሚክ ሰራተኛ የት/ት ክፍል ኃላፊ ስምና ፊርማ ----------
- ስአካዳሚክ ሰራተኛ ኮሌጅ ዲን/ ኃላፊ ስምናፊርማ ----------

ተ.ቁ	የስራክፍሎች	ሙሉ ስም	ፊርማ	ቀን
1	ሰርኩሌሽን አስተባባሪ 1			
2	ሰርኩሌሽን አስተባባሪ 2			
3	ሰርኩሌሽን አስተባባሪ 3			
4	የቤተ-መፀሐፍትና ኢ/ማዕከል ዳይሬክተር			
5	የሒሳብ ሰራተኛ			
6	ዋና ገንዘብ ያዥ			
7	የፋይናንስ ዳይሬክቶሬት			
8	የቋሚ እቃ ግምጃ ቤት ሰራተኛ (ኤሌክትሮኒክስ)			



9	የቋሚ እቃ ግምጃቤ ትሰራተኛ (ፈርኒቸር)			
10	የቋሚ እቃ ግምጃቤትሰራተኛ (ቋሚ/አላቂ)			
11	የግዥና ን/አስ/ዳይሬክቶሬት			
12	የመምህራንና ሰራተኞች ገ/ቁ/አ/ማ/ሒሳብ ሹም			
13	ሬጅስትራር ጽ/ቤት			
14	የቤቶች አስተዳደር አስተባባሪ			
15	ዩኒቨርሲቲ ኢንዱስትሪ ትስስር ዳሬክቶሬት			
16	ማህበረሰብ አገልግሎት ዳይሬክቶሬት			
17	ሀገር በቀል እውቀት ዳይሬክቶሬት			
18	ምርምር ዳይሬክቶሬት			
19	ምርምርና ማ/አገ/ም/ፕ/ጽ/ቤት			
20	አይሲቲ ዳይሬክቶሬት			
21	አካዳሚ ክፍሎች ም/ፕ/ጽ/ቤት			
22	አስተዳደርና ኮ/ማ/ም/ፕ/ጽ/ቤት			
23	የሪከርድና ማህደር ሰራተኛ (ክስ ፣ዋስትና፣ ከእዳ ወዘተ ነፃ መሆኑን ያረጋገጠው)			
24	የሰው ሀብት አስ/ል/ዳይሬክቶሬት			

ማሳሰቢያ፣

- ይህ ቅጽ በሁለት ቅጂ ተሞልቶ የሰው ሀ/አስ/ል/ዳይሬክቶሬት ክፍል ፈርሞ ከታተመ በኋላ አንድ ቅጽ ለሰራተኛው ሲሰጥ ሌላኛው ከሰራተኛው ማህደር ጋር ይያያዛል።
- ክሊራንስ ከተራ ቁጥር 1—24 ባለው ባተራ ቁጥር በቅደም ተከተል በኃላፊዎች /በሚመለከተው የስራ ክፍል ይፈረማ።

Figure 2. Form signed for Academic staff clearance process.

Wolkite University
ወልቂጤ ዩንቨርሲቲ

ቁጥር ----------
ቀን ----------

Human resource M/D/Directorate
የሰዉ ሀብት አስተዳደር ልማት ዳይሬክቶሬት

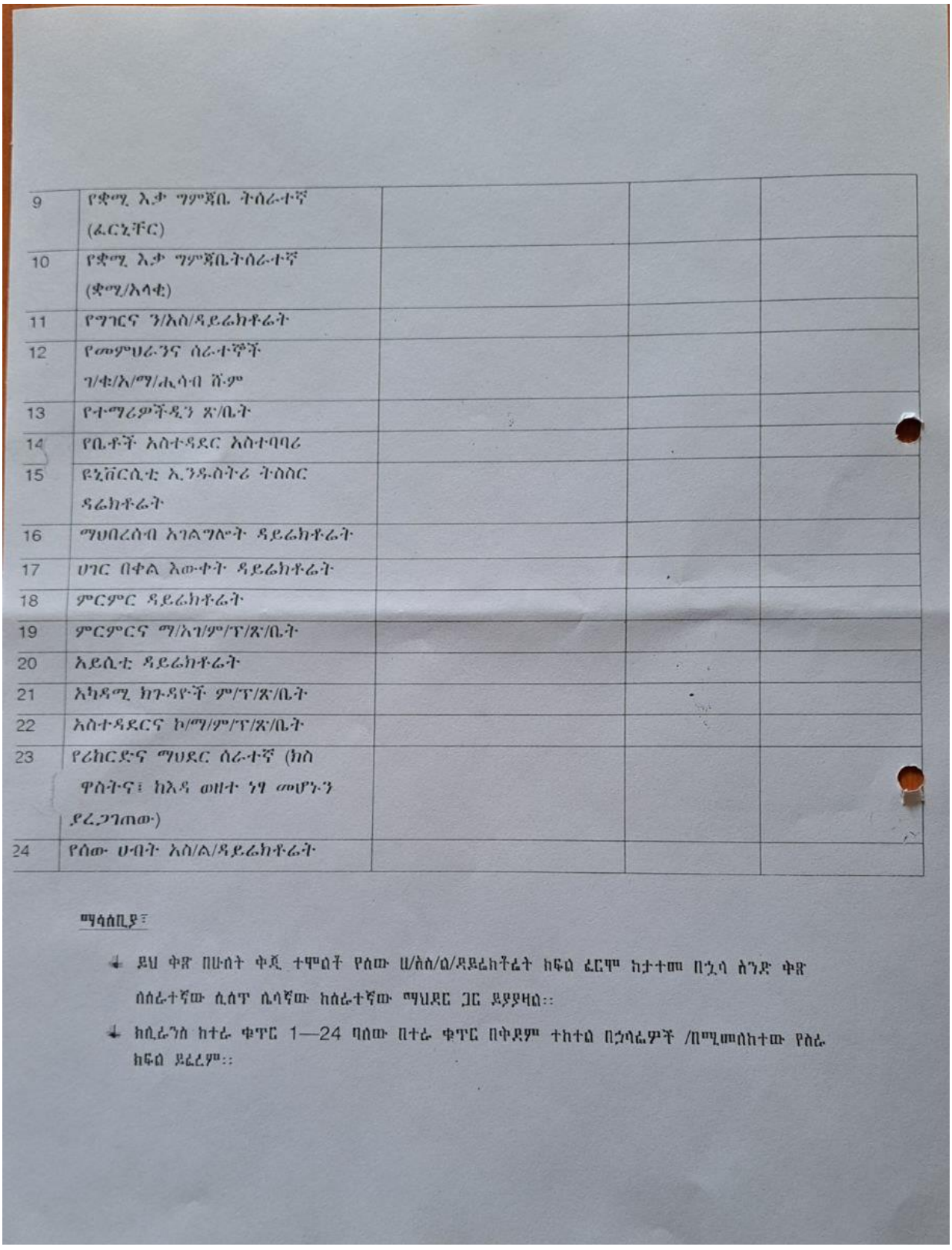
የአስተዳደር ሰራተኛ ንብረት ማስረከቢያ ቅጽ/ክሊራንስ/

ሀ/ የዚህ ንብረት ማስረከቢያ ቅጽ አገልግሎት እንድ የየኒቨርሲቲው አባል፡

5. ለንብረት ቁጥጥር ሲባል በአዲያለውንየየኒቨርሲቲውንንብረትማስረከብቢናርበት
6. ከሌሎች ከንብረትና ገንዘብ አዳ ጋር በተገናኘ መረጃ ሲፈለግ
7. ለተመሰነ ጊዜስት/ት ወይም ለምርምር ከመደበኛ ሰራው የመንግስት ንብረት በእድ አለመኖረን ማረጋገጫ
8. ሰራተኛው ስራውን በሚለቅበት ወቅት ክሊራንስ ከመሰጠቱ በፊት የመንግስት ንብረትና ገንዘብ ያስረከቡ መሆኑን ማረጋገጫ

ለ/ የሰራተኛ ስም ከነአያት ----------
የስራ መደቡ መጠሪያ ----------የመ/ቁጥር ----------የሰራ ደረጃ --------ደመወዝ ------
የስራክፍል ---------- የሰራተኛው ፊርማ ----------
ሐ/ የገንደስ ንብረትና ገንዘብ ቢናር ተጠያቂ የሚሆን ዋስ ስም ከነአያት ----------
የሚሰራበት ስራ ክፍል ---------- የዋስ ፊርማ ----------
መ/ ከስራ ገበታ የሚለየበት ምክንያት----------
ቀን ----------
ሠ/ቅጹ እንዲሞላ የፈቀደው ሃላፊ
ስለስተዳደር ሰራተኛ የዳይሬክተሩ/ሯ ስምና ፊርማ ----------

ተ.ቁ	የስራክፍሎች	ሙሉ ስም	ፊርማ	ቀን
1	ሰርኩሌሽን አስተባባሪ 1			
2	ሰርኩሌሽን አስተባባሪ 2			
3	ሰርኩሌሽን አስተባባሪ 3			
4	የቤተ-መፀሐፍትና ኢ/ማዕከል ዳይሬክተር			
5	የሒሳብ ሰራተኛ			
6	ዋና ገንዘብ ያዥ			
7	የፋይናንስ ዳይሬክቶሬት			
8	የቋሚ እቃ ግምጃ ቤት ሰራተኛ (ኤሌክትሮኒክስ)			



9	የቋሚ እቃ ግምጃቤ ትሰራተኛ (ፈርኒቸር)			
10	የቋሚ እቃ ግምጃቤትሰራተኛ (ቋሚ/አላቂ)			
11	የግገርና ን/አስ/ዳይሬክቶሬት			
12	የመምህራንና ሰራተኞች ገ/ቁ/አ/ማ/ሒሳብ ሹም			
13	የተማሪዎችዲን ጽ/ቤት			
14	የቤቶች አስተዳደር አስተባባሪ			
15	ዩኒቨርሲቲ ኢንዱስትሪ ትስስር ዳሬክቶሬት			
16	ማህበረሰብ አገልግሎት ዳይሬክቶሬት			
17	ሀገር በቀል እውቀት ዳይሬክቶሬት			
18	ምርምር ዳይሬክቶሬት			
19	ምርምርና ማ/አገ/ም/ፕ/ጽ/ቤት			
20	አይሲቲ ዳይሬክቶሬት			
21	አካዳሚ ክፍሳዮች ም/ፕ/ጽ/ቤት			
22	አስተዳደርና ኮ/ማ/ም/ፕ/ጽ/ቤት			
23	የሪክርድና ማህደር ሰራተኛ (ክስ ዋስትና፣ ከእዳ ወዘተ ነፃ መሆኑን ያረጋገጠው)			
24	የሰው ሀብት አስ/ል/ዳይሬክቶሬት			

ማሳሰቢያ፣

- ይህ ቅጽ በሁለት ቅጂ ተሞልቶ የሰው ሀ/አስ/ል/ዳይሬክትሬት ክፍል ፈርሞ ከታተመ በኋላ አንድ ቅጽ ሰሰራተኛው ሲሰጥ ሲላኛው ከሰራተኛው ማህደር ጋር ይያያዛል።
- ክሲራንስ ከተራ ቁጥር 1—24 ባለው በተራ ቁጥር በቅደም ተከተል በኃላፊዎች /በሚመለከተው የስራ ክፍል ይፈረማ።

Figure 3. Form signed for Admin staff clearance process.

Chapter 3

3.Proposed system

3.1 Introduction

Through meticulous observation and user interviews, the project team identified inherent challenges within the manual-based clearance system. In response to these challenges, the project team proposes the development of an automated clearance system. The primary objective of this proposed system is to mitigate the existing problems and weaknesses observed in the current clearance process. The envisioned automated clearance system will introduce a centralized approach, allowing members of the university community to obtain their clearances seamlessly from a single location, eliminating the need to navigate through various offices.

This proposed system aims to streamline resource control mechanisms, fostering an environment of efficiency and interconnectedness between relevant university offices. The envisioned solution will facilitate the clearance request process by automating the verification and approval steps. When a clearance request is submitted, the system will conduct a thorough check of the information provided, ensuring accuracy and completeness. Approved clearance forms will be displayed to the requestor, and the system will systematically store the approved clearances for future reference.

In essence, the proposed system is poised to address the current challenges and resource consumption associated with the university's manual clearance activities. By transitioning to an automated system, the university community can anticipate a more efficient, centralized, and communication-enhanced clearance process, marking a significant advancement from the limitations of the existing manual system.

3.2 Functional requirement

The proposed automated clearance system incorporates a comprehensive set of functional requirements to ensure a seamless and efficient user experience. For students, the system mandates the establishment of user accounts, granting access to clearance forms, and facilitating the request for approval and issuance of clearance certificates. Similarly, staff members follow a

parallel process, engaging with their own user accounts to navigate clearance forms and obtain necessary approvals. The administrative role encompasses the registration and management of user accounts, the assignment of roles to system stakeholders, and the facilitation of user account updates. Officers hold pivotal responsibilities in the system, with the ability to either approve or reject requested clearances. Additionally, the system supports on-screen clearance printing for students and provides officers with a comprehensive view of clearance statuses. The system administrators role further includes the capability to generate and view reports. This robust set of functional requirements collectively contributes to the envisioned efficiency and effectiveness of the proposed automated clearance system. All of the functional requirements are listed as follows:

- ✓ The student should have an account.
- ✓ The student should access the clearance form.
- ✓ The student should request approval.
- ✓ The student should generate the certificate of clearance.
- ✓ The staff should have an account.
- ✓ The staff should access the clearance form.
- ✓ The staff should request approval.
- ✓ The staff should generate the certificate of clearance.
- ✓ The admin should be able to register new users.
- ✓ The admin should be able to manage user accounts.
- ✓ The admin should be able to grant roles to the stake holders of the system.
- ✓ The officers should be able to reject requested clearance.
- ✓ The officers should be able to approve requested clearance.
- ✓ The system should allow users to update their own account.
- ✓ The system should allow students to request for clearance approval.
- ✓ The system should allow students to be able to print clearance on-screen.
- ✓ The system should allow officers to view clearance status.
- ✓ The system should allow admin to view generated report.
- ✓ The system should allow admin to view generated report.

3.3. Non-functional requirement

3.3.1. User Interface and Human Factors

Given that the main stakeholders of the system include university students, staff, and officers, a critical non-functional requirement is to ensure an intuitive user experience (UX) and user interface (UI). Many users may lack semantic and syntactic knowledge of web applications, necessitating a user-friendly design. To achieve this, the proposed system will adhere to UX/UI design principles prioritized in the following order:

1. **Use of Professional Templates:**
   - The system will implement professional templates created by seasoned UI/UX designers and developers to enhance the overall aesthetic and functionality of the interface.
   - **Reasoning:**
     - **Accessibility:** Professionally designed templates ensure accessibility for users with varying levels of technical expertise.
     - **Consistency:** Standardized templates contribute to a consistent and cohesive user experience throughout the system.
     - **Usability:** Expertly crafted templates prioritize usability, simplifying navigation and interaction for all users.
2. **Responsive Design:**
   - The system will incorporate a responsive design approach, ensuring optimal functionality across various devices such as desktops, tablets, and mobile phones.
   - **Reasoning:**
     - **Adaptability:** Responsive design guarantees an adaptable interface, accommodating users accessing the system from different devices.
     - **Enhanced Accessibility:** Users can interact with the system seamlessly, irrespective of the device used, enhancing overall accessibility.

3. **Intuitive Navigation:**

   - The system will feature intuitive navigation structures, minimizing the learning curve for users and enhancing the overall ease of interaction.

   - **Reasoning:**

     - **User-Friendly:** Intuitive navigation promotes a user-friendly environment, reducing the likelihood of user errors and frustration.

     - **Efficiency:** Users can swiftly locate and engage with system features, contributing to an efficient workflow.

By incorporating these design principles, the proposed system aims to create an inclusive, accessible, and user-centric environment, ensuring a positive and productive experience for all stakeholders within the university community.

3.3.2. Hardware Consideration

An essential non-functional requirement for the proposed system is seamless compatibility and responsiveness across popular mobile and desktop web browsers. The system must ensure optimal performance without any malfunctioning, addressing the varying hardware considerations of users. This design approach is particularly crucial as it accommodates the diverse preferences and needs of the university community.

1. **Compatibility Across Browsers:**

   - The proposed system will be meticulously designed to function smoothly across a spectrum of popular mobile web browsers (such as Chrome, Safari, Firefox) and desktop web browsers (like Chrome, Firefox, Edge). This ensures users can access and interact with the system regardless of their chosen browser, promoting inclusivity and flexibility in usage.

2. **Responsiveness Across Devices:**

   - The system will prioritize responsiveness to guarantee a seamless user experience on both mobile devices and desktops. By adapting to different screen sizes and

resolutions, the system ensures consistent functionality and user interface quality across a variety of devices.

3. **Mobile-Friendly Design:**

    - Recognizing the prevalence of mobile device usage, especially among students, the proposed system will be optimized for mobile web browsers. This allows users to conveniently process clearance approvals using their mobile devices, eliminating the need for a dedicated desktop application.

By addressing hardware considerations in this manner, the proposed system aligns with contemporary user habits and expectations. The emphasis on compatibility and responsiveness fosters a user-centric approach, enhancing accessibility and usability for all members of the university community, regardless of their preferred device.

3.3.3. Security Issues

Security is of paramount importance for the proposed system, necessitating robust measures to safeguard user data and system integrity. The system will employ a multifaceted security strategy encompassing user identification, categorization, and authentication, coupled with stringent authorization mechanisms. This approach aims to prevent unauthorized access, protect against potential attacks, and ensure the overall resilience of the system.

Authentication and Authorization Mechanisms:

1. **User Identification and Categorization:**

    - The system will implement a meticulous user identification process, categorizing users based on their roles and responsibilities within the university community. This ensures a granular approach to privileges and access control.

2. **Authentication Protocols:**

    - NextAuth, a state-of-the-art authentication library, will be utilized to fortify the system's authentication process. NextAuth facilitates seamless integration with various identity providers, ensuring secure and reliable user authentication.

Significance of NextAuth:

- **Adaptive Authentication:** NextAuth supports adaptive authentication, dynamically adjusting the level of authentication based on contextual factors such as user behavior and location. This enhances security by identifying and responding to potential threats.
- **Multi-Factor Authentication (MFA):** NextAuth enables the implementation of MFA, an additional layer of security that requires users to provide multiple forms of identification. This significantly reduces the risk of unauthorized access.
- **Social Authentication Integration:** The library supports integration with social authentication providers, offering users the convenience of using their existing social media credentials for authentication while maintaining robust security standards.
- **Session Management:** NextAuth includes robust session management features, allowing for secure and efficient handling of user sessions. This ensures that user interactions with the system remain secure and uninterrupted.

By adopting NextAuth, the proposed system not only adheres to industry best practices but also leverages advanced features to enhance the overall security posture. The use of NextAuth aligns with the system's commitment to safeguarding user data, thwarting potential security threats, and providing a trustworthy platform for all stakeholders.

3.3.4. Performance Consideration

Efficient performance is a fundamental non-functional requirement for the proposed system, given its dynamic user base with diverse needs. The system is expected to promptly handle and process user requests, ensuring a seamless experience for all users. To achieve optimal performance, the system will leverage Next.js, a powerful React framework known for its efficiency, speed, and scalability.

Performance Optimization Strategies:

1. **Rapid Request Processing:**
    - Next.js's server-side rendering (SSR) capabilities will be harnessed to expedite the processing of users' requests. SSR ensures that content is pre-rendered on the

server, reducing the time it takes to load pages and enhancing overall responsiveness.

2. **Efficient Resource Utilization:**
    - The system will be configured to efficiently utilize resources, responding quickly to user interactions and minimizing the consumption of system resources. This approach aims to provide users with a fluid and uninterrupted experience.
3. **Compact System Footprint:**
    - Next.js's lightweight and minimalistic approach to system architecture will contribute to a compact system footprint. This ensures that the system occupies minimal space while delivering robust performance, enhancing overall efficiency.

3.3.5. Error Handling and Validation

Ensuring robust error handling and fault tolerance is imperative for the proposed system, acknowledging that users may encounter errors and the system itself may face runtime failures. The system will implement effective exception handling methods to address both user-generated errors and internal system issues.

User-Generated Error Handling:

1. **Exception Handling for User Errors:**
    - The system will incorporate proper exception handling methods to address errors resulting from user actions. This includes validating user inputs and providing informative error messages to guide users in rectifying their mistakes.
2. **Input Fault Reduction Strategies:**
    - To mitigate input faults, the system will implement the following measures:

- **Data Verification Before Processing:** Users will have the opportunity to review and confirm the accuracy of information before any creation, processing, or update occurs.

- **User-Friendly Data Reentry:** In cases of incorrect data entry, the system will prompt users to reenter the data in the correct format, minimizing input errors.
- **Error Management in Database Interactions:** The system's database interactions and input accepter fields will be equipped with error management mechanisms to handle and rectify issues seamlessly.

Yup for Form Validation:

- Form validation will be facilitated through Yup, a robust JavaScript schema validation library. Yup ensures that data entered into forms adheres to predefined rules, enhancing the accuracy and integrity of user inputs.

Proactive Issue Recognition:

- In the event of an issue, the system will proactively recognize and communicate the error to the user, enabling them to make necessary corrections. This approach emphasizes user empowerment and prevents system shutdowns due to manageable errors.

By adopting a proactive and user-centric approach to error handling, the proposed system aims to enhance the overall user experience, promote data accuracy, and ensure continued functionality even in the face of runtime challenges.

3.3.6. Quality Issues

Given its crucial role in the university's clearance procedure and resource control, the proposed system places a premium on precision, reliability, and robustness. These quality attributes are foundational to the system's effectiveness in managing user requests and providing accurate information.

Quality Objectives:

1. **Reliability:**
    - The system is designed to reliably retrieve and display only user-requested data. Users can place full trust in the system, confident that the information presented is both accurate and dependable. The system’s reliability achieved through

meticulous data retrieval mechanisms and a commitment to displaying information precisely as requested by the user.

2. **Robustness:**

    - WKUSCMS is engineered to be robust, capable of identifying errors and guiding users toward the appropriate corrective actions. This emphasis on robustness ensures a resilient and user-friendly experience. The system leverages error-catching mechanisms to identify and address issues promptly, prioritizing user guidance for seamless interaction.

3. **Security Issues:**

    - Security is paramount, with the system incorporating a secure login page that restricts access to only authorized users. This stringent access control mechanism prevents unauthorized individuals from infiltrating the system. Users gain access to the system by registering and logging in, with access rights determined by their roles within the university. Non-functional requirements, including security protocols, remain integral components of the system's visible yet supportive features.

    ➢ **User Authentication and Authorization:**

        - Every user undergoes a registration process and subsequent login to access the clearance form, ensuring that only authenticated users can interact with the system.

        - Access rights are meticulously managed, aligning with each user's specific roles and responsibilities within the university community.

By emphasizing reliability, robustness, and security, the proposed system aims to instill user confidence, facilitate error-free interactions, and uphold the integrity of the clearance procedure within the university's resource control framework.

3.3.7. Backup and Recovery

Automated backups of the system data will be conducted regularly, stored securely to mitigate data loss. In case of system failure, a streamlined recovery process will restore the system to a recent state. The chosen database's inherent reliability ensures efficient data recovery, enhancing the overall robustness of the system.

3.3.8. Physical Environment

The system is designed to operate seamlessly in diverse physical environments, accommodating users across various locations and devices. Its adaptability ensures optimal performance whether accessed from mobile web browsers or desktops, promoting accessibility and user convenience.

3.3.9 Documentation

Comprehensive documentation will be provided with the new system. Because of this, other developers out of our team will have the necessary resources for efficient system understanding, fosters ease of maintenance, and facilitates a smooth transition to and continued use of the updated system.

Chapter 4

4. System Analysis

4.1. System Model

In this pivotal chapter, the system model takes shape through a meticulous exploration of key tasks. A comprehensive use case model is crafted, detailing the sequence of events for each use case. Object-oriented diagrams, including the system model use case diagram and object model class diagram, are instrumental in structuring the envisioned system. Dynamic models such as state charts, sequence diagrams, and activity diagrams are strategically employed, providing a detailed and insightful portrayal of the system's functionality. This analytical process is underpinned by the systematic creation of a user interface prototype, ensuring a holistic and well-structured approach to system analysis.

4.1.1. Use case Model

- The use case model unfolds with pivotal actors defining the operational dynamics:

1. **Users:** This encompassing actor category features three distinct roles, each playing a vital part in the system's functionality:
    - **Student:** Initiates clearance requests, seeks certificate issuance, and interacts with the system for various academic-related processes.
    - **Staff:** Engages in multifaceted clearance procedures, contributing to the system's comprehensive workflow. The staff role further diversifies into two subcategories:
        - **Normal Staff:** Undertakes standard clearance processes, ensuring the seamless flow of routine tasks.
        - **Staff with Approval Permission:** Assumes additional responsibilities by holding the authority to approve clearance requests, adding a layer of authorization to the workflow.
2. **System Administrator:** A pivotal role responsible for system management and control, ensuring the overall integrity and efficiency of the proposed system.

This hierarchical actor structure enhances clarity and granularity, allowing for a nuanced representation of user roles within the system. It ensures that each actor, from students to staff with distinct permissions, is precisely defined, contributing to a detailed and effective use case model.

List of Use cases found from the functional Requirements**.**

- Use cases of the System Administrator
  - ✓ Manage office account
  - ✓ Manage user account
  - ✓ Post announcement
  - ✓ Replay feedback
  - ✓ Generate Report
- Use cases of the service consumer (student and staff)
  - ✓ Request clearance approval
  - ✓ View clearance status
  - ✓ Send feedback
  - ✓ View announcement
  - ✓ Update Profile
  - ✓ Request certificate of clearance
- Use cases of the staffs with approval permissions
  - ✓ Approve clearance request
  - ✓ Reject clearance request
  - ✓ Send reasons of rejection
  - ✓ Update Profile

❖ All of the actors that are listed above includes the Login for Logout usecase.

4.1.2. Use case Diagram

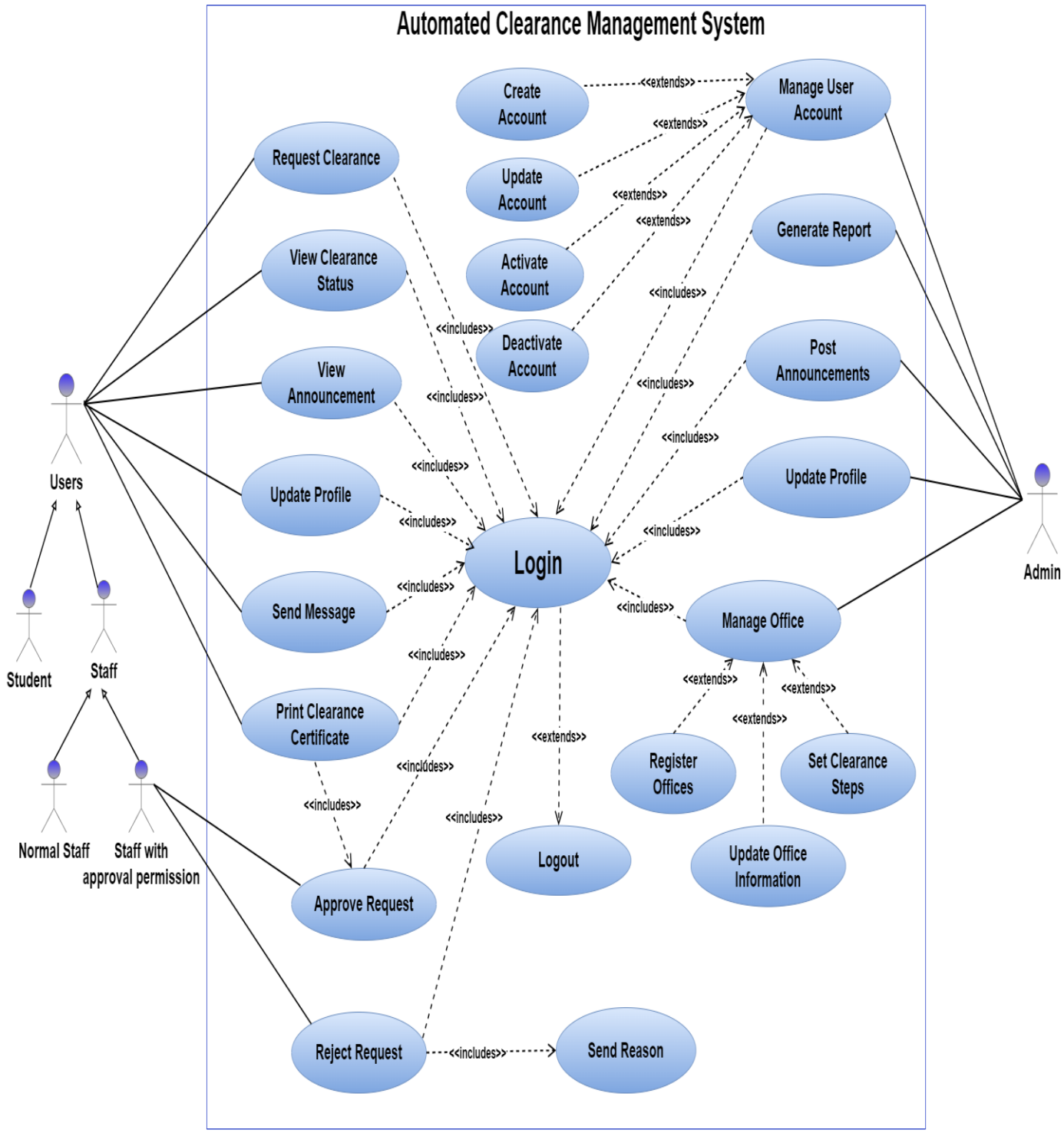


Figure 4. Usecase diagram

4.1.3. Use Case Description

Table 1. Login use case description

Use case name	Login
Participating actor	All system user
Description	Any user who wants to use the system's functionality must first log in and be authenticated and authorized.
Entry condition	The user must already be registered (the user must have username, password and account role).
Flow of event	1. The user opens the system. 2. The user clicks the login link. 3. The system displays the login page 4. The user enters his or her personal information (Username and password) 5. The user clicks the login button. 6. The system verifies the username and password. 7. The system takes the user to his/her 8. page. Use case end.
Alternative flow of event	6.1. If the user inserted wrong username and password 6.2. the system toasts error and try again message.
Post condition	Users logs to the system.

Table 2. Create user account use case description

Use case name	Create user account
Participating actor	Admin
Description	When an admin needs to create a new user account, this use case is helpful.
Entry condition	The admin should login to the system.
Flow of event	1. Admin page displayed (system response). 2. The admin choose the add button on Manage account in the menu bar (Actor action). 3. The system shows the form for adding (system response). 4. The Admin fills the required information (Actor action). 5. Click create account (Actor action). 6. Account created toast will be displayed (system response). Use case ends.
Alternative flow of event	Step 5.1. if admin inserted already existing id, the system displays “User already exist ” message. Step 5.2 If user enters wrong information the system display message in order to correct wrong. information.
Post condition	A new user account is created.

Table 3. Deactivate account use case description.

Use case name	Deactivate account
Participating actor	Admin
Description	This use case helps the admin in deleting the user account if it is no longer required.
Entry condition	The admin login to the system, the account exists.
Flow of event	1. The Admin selects the account that he wants to deactivate. 2. A popup will be displayed that asks if he approves to deactivate. 3. An admin confirms by clicking the delete button. 4. The account will be deactivated. use case end.
Alternative flow of event	Step 1-2 remain the same. Step 3. if the admin doesn't confirm it will leave stop this process.
Post condition	The account is deleted.

Table 4. update profile use case description

Use case name	Update profile
Participating actors	Student, staffs and admin
Description	The above actors can update each of the information of themselves.
Precondition	user login to the system.
Basic course of action	1. User selects Manage account link from navbar dropdown. 2. The system display information of that account. 3. User makes necessary modification and click Update

	button. 4. The system asks for conformation. 5. User click ok button. 6. The system saves the change to that account. 7. The system displays an acknowledgement successfully updating the account. use case ends
Post condition	Save the change to the account.

Table 5. Request clearance use case description

Use case name	Request clearance
Participating actor	Students, staffs
Description	The actor sends a clearance request to the staff (who has approval permission) due to different clearance reasons.
Entry condition	There must not be other request triggered by the actor before and which is still processing.
Flow of event	1. The actor must select a specific reason for his clearance request. 2. The actor clicks on request clearance link. 3. The system must send the request to the concerning office respectively as it steps. 4. The system display success message for the request.
Alternative flow of event	If the user already sends the request the send request button will not be triggered until he finishes the approval process.
Post condition	Clearance successfully requested.

Table 6. view clearance status use case description

Use case name	view clearance status
Participating actor	Students, staffs
Description	The actor tracks a clearance request status after once sends the request.
Entry condition	The actor should request a clearance before.
Flow of event	1. The actor should click the clearnce status tab. 2. The actor tracks how his progress is going. Use case ends.

Table 7. post announcements use case description

Use case name	Post announcement	
Participating actors	Admin	
Description	The admin can post the announcements for the users.	
Precondition	To post announcements, the actor must be logged in to the system.	
Basic course of action	Actor action	System response
	1. The admin clicks the post icon or button from the home of the web sidebar. 3. The users posts its announcements to the users.	2. The system displays a posting field with actions. 4. The system display message for posting successfully. Then the use case ends.
Post condition	admin can successfully post their announcements.	

Table 8. View announcements use case description.

Use case name	view announcement	
Participating actors	Students and staffs	
Description	All the above actors of the system can see the announcements that are posted for them.	
Precondition	To view announcements the actor must be logged in to the system.	
Basic course of action	Actor action	System response
	2. The users views its announcement that are sent from posted by the admin. Then the use case ends.	1. The system displays a announcement that are posted by others.
Post condition	User can successfully view the posts that are posted by the admin.	

Table 9. Send messages use case description

Use case name	send feedback
Participating actors	Students, staffs and admin
Description	All actors of the system may have need to chat during clearance process for example there will be messaging for clarifying the reason of rejection.
Precondition	User must login on the web page.
Basic course of action	Actor action System response

	1. The user selects the user to whom the message will be send. 2.The user writes the message. 2. The user clicks the send button. 4.The system displays message for successful message delivery. Use case end.
Alternate course of action	If the user wants not to send the comment, the system returns to basic flow web home page.
Post condition	User can successfully send their feedback and comment.

Table 10. send feedback use case description.

Use case name	send feedback	
Participating actors	Students, staffs	
Description	All actors of the system may have some comment and feedback during clearance process about the system at this time they can send feedback to the other system administrator.	
Precondition	User must login on the web page.	
Basic course of action	Actor action	System response

	1. The user clicks the send feedback button from the home page of the web. 3.The users sends its feedback for the admins.	2. The system displays a messaging field with actions. 4.The system displays message for successful message delivery. Use case end.
Alternate course of action	If the user wants not to send the comment, the system returns to basic flow web home page.	
Post condition	User can successfully send their feedback and comment.	

Table 11. View feedback use case description

Use case name	view feedback	
Participating actors	admin	
Description	Administrator of the system can see the feedbacks or messages that are written for them.	
Precondition	To view feedback the actor must be logged in to the system.	
Basic course of action	Actor action	System response
	1. The user clicks the message icon or button from the home of the web. 3. The users views its feedbacks or messages that are sent from others. Then the use case ends.	2. The system displays a message that are sent by others.
Post condition	admin can successfully view the feedbacks.	

Table 12. set clearance step use case description

Use case name	set clearance step
Participating Actors	Admins
Description	The system administrators shall organize the steps of the clearance. This feature is add for making the clearance process dynamic.
Pre-condition	The offices must be registered.
Basic course of action	1. The admin opens manage steps section by clicking its link on the sidebar. 2. The admins sets the step of offices for the clearance process. 3. Then saves the organized step.
Alternate course of action	If the admin wants to left setting up the step he can left it.
Post conditions	Will allow the requester to continue the next step of clearance.

Table 13. Approve request use case description.

Use case name	Approve request
Participating Actors	Staffs (with approval permission)
Description	When the students and staffs request clearance approval to an office the staff(who has approval permission) will check manually and approves if the actor returns all borrowed Items.
Pre-condition	The students or staffs should request clearance Approval.

Basic course of action	1. selects one or multiple requesters. 2. Click the approve button. 3. System displays a toast for successful approval of the request.
Alternate course of action	If the approver wants to stop approval he can left it.
Post conditions	Will allow the requester to continue the next step of clearance.

Table 14. Reject request use case description.

Use case name	Reject request
Participating Actors	Staffs (with approval permission)
Description	When the students and staffs request clearance approval to an office the staff(who has approval permission) will check manually and rejects if the actor hasn't returne borrowed Item and will tell them the reason.
Pre-condition	The student should request clearance Approval.
Basic course of action	5. selects one aro multiple requesters. 6. Click the reject button. 7. Rejection confirmation popup displayed. 8. Confirm to reject the request. 9. Write and send basic reason of rejection for the requester.
Alternate course of action	If the approver wants to stop rejection he can left it.
Post conditions	Will not allow the requester to continue the next step of

	clearance.

Table 15. Request certificate of clearance usecase description

Use case name	Request certificate of clearance
Participating actors	Student and staff
Description	This use case allows above actors to generate their certificate.
Precondition	the actors must finish all finish the whole clearance process.
Basic course of action	1. The actors selects the generate certificate. 2. a system will start to download their certificate of clearance. Use case ends.
Post condition	System stores the certificate dertails into database.

Table 16. Generate report usecase description

Use case name	Generate report
Participating actors	admin
Description	This use case allows a system admin to generate report.
Precondition	All information regarding with the process must be gathered.
Basic course of action	1. The admin selects the reports link on the sidebar. 2. a system admin generates reports by clicking gr=enerate report button. Use case ends.
Post condition	System stores report into database.

Table 17. Logout use case description

Use case name	Logout
Participating actor	All system users
Description	Any user who wants to logout from the system the user clicks the logout button.
Entry condition	The user must be login to the system.
Flow of event	1.The system displays the logout button. 2.The user clicks the logout button.
Alternative flow of event	
Post condition	The user his home page and redirected to the landing page.

4.1.4. Use case scenario

WKUCMS has many different scenarios that are divided in to participating actors and user groups. In this specific case we identify the following scenarios that are considered as main functionality of the system.

1. Clearance Request Scenario

Participating Actor: Sarah (Student)

Entry Condition: Sarah has not submitted any previous clearance requests that are still in process.

Flow of Events:

1. **Selection of Reason:**
    - ✓ Sarah logs into the university's online portal.
    - ✓ She navigates to the "Clearance Requests" section.
    - ✓ A list of clearance reasons is presented, and Sarah selects "End of year" as the reason for her request.

2. **Initiating Clearance Request:**
    - ✓ Sarah clicks on the "Request Clearance" link.
3. **System Submission:**
    - ✓ The system processes her request and identifies the appropriate office for Clearance.
    - ✓ The request is automatically sent to her corresponding department head for approval.
4. **Success Message:**
    - ✓ Upon successful submission, the system displays a confirmation message: " Clearance Request Submitted Successfully."

Alternative Flow of Events:

- ✓ If Sarah had already sent a Clearance request that is still pending approval:
    - ✓ The system detects the existing request and prevents Sarah from triggering a new request until the previous one is resolved.
    - ✓ A message is displayed, informing Sarah that she cannot submit a new request until the previous one is processed.

Post Condition: Sarah's Clearance request has been successfully submitted, and it is now in the approval process.

2. Approval request scenario

Participating Actor: Dr. Alemu (Department Head)

Pre-condition: Students or staff have submitted a clearance approval request.

Basic Course of Action:

1. **Login and Navigation:**

- ✓ Dr. Alemu logs into the wkucms portal.
- ✓ As the Department Head, he has access to the "Clearance Approval" section.

2. **Selection of Requesters:**
   - ✓ Dr. Alemu reviews the list of clearance approval requests pending in the system.
   - ✓ He selects one or multiple requesters from the list.
3. **Approval Process:**
   - ✓ After selecting the requesters, Dr. Alemu clicks on the "Approve" button.
4. **Manual Check:**
   - ✓ Dr. Alemu manually checks each selected requester's account to ensure they have returned all borrowed items, considering the department-specific requirements.
5. **Approve Button Clicked:**
   - ✓ Satisfied with the clearance status, Dr. Alemu clicks the "Approve" button.
6. **Success Toast:**
   - ✓ The system processes the approval and displays a toast message: "Clearance Request Approved Successfully."

Alternate Course of Action:

- ✓ If, during the manual check, Dr. Alemu decides not to approve the request:
  - ✓ He chooses not to click the "Approve" button.
  - ✓ The system retains the request in its pending state.

Post Conditions:

- ✓ The approved clearance request is now marked as successfully processed.
- ✓ The system will send the request to the next approver.

- ✓ The requesters can proceed can track of the clearance process, knowing their request has been approved by Dr. Alemu, the Department Head.

3. Request Certificate of Clearance scenario

Participating Actor: Abel (Staff)

Precondition: Abel has completed the entire clearance process.

Basic Course of Action:

1. **Completion of Clearance Process:**
    - ✓ Abel successfully completes all steps of the clearance process, including approval from the department head.
2. **Generate Certificate:**
    - ✓ Abel logs into the wkucms's online portal.
    - ✓ He navigates to the "Certificate of Clearance" section.
3. **Certificate Generation:**
    - ✓ Abel clicks on the "Generate Certificate" button.
4. **System Processing:**
    - ✓ The system, recognizing that Abel has completed the entire clearance process, starts to generate his Certificate of Clearance.
5. **Download Certificate:**
    - ✓ A download prompt appears, and Abel downloads the generated Certificate of Clearance to his device.
6. **Use Case Ends:**
    - ✓ The use case concludes as Abel successfully downloads his Certificate of Clearance.

Post Condition:

- ✓ The details of Abel's Certificate of Clearance are stored in the system's database.

Note: This scenario assumes that Abel is a staff member who has successfully completed the clearance process, allowing him to generate and download his Certificate of Clearance.

4. Report Generation Scenario

Participating Actor: Jemal (System Administrator)

Precondition: All information related to the processes has been gathered and is available for reporting.

Basic Course of Action:

1. **Log in and Navigation:**
    - ✓ Jemal, the System Administrator, logs into the system with his administrative credentials.
    - ✓ He navigates to the "Reports" link located in the sidebar.
2. **Selection of Report Type:**
    - ✓ Jemal identifies the type of report he needs to generate based on the available options.
    - ✓ For example, he might choose a "Clearance Status Report" to monitor the overall clearance process.
3. **Generate Report:**
    - ✓ Jemal clicks on the "Generate Report" button.
4. **System Processing:**
    - ✓ The system processes the request, compiling the relevant data and generating the selected report.

5. **View and Validate Report:**

    - ✓ Once the report is generated, Jemal reviews it to ensure it contains accurate and up-to-date information.

6. **Use Case Ends:**

    - The use case concludes as Jemal successfully generates the report and validates its contents.

Post Condition:

- ✓ The generated report is stored in the system's database.
- ✓ Jemal, as the System Administrator, now has access to the report for analysis or distribution as needed.

4.2. Object model

In this section, the Object Model materializes, capturing the essence of the system's structure through a concise yet informative Class Diagram. The Class Diagram offers a high-level overview of the key entities and their relationships within the system. Additionally, a data dictionary is included to provide brief descriptions of the attributes associated with each class. This snapshot of the system's data structure facilitates a quick understanding of the fundamental components and their interconnections. The Object Model contributes to a holistic representation, ensuring a robust foundation for system analysis and design.

4.2.1. Class diagram

Figure 5. Class diagram

4.2.2. Data dictionary

Table 18. Data Dictionary for User

Field Name	Data Type	Description
First Name	String	First name of an individual
Middle Name	String	middle name of an individual
Last Name	String	Surname of an individual
Id	Object id	Unique identifier for the record or entity
User Id	String	University's Id that is given for members
College	String	User's college
Password	String	Password associated with the individual's account
Department	String	User's department
Image URL	String	Accounts profile picture image url
Year	String	Year of students
Role Id	String	Position of the individual within a system
Permission	String	Permissions granted to the individual within a system

Table 19. Data Dictionary for Notification

Field Name	Data Type	Description
Id	Object id	A unique identifier for the notification record
Notification Type	String	Type or category of the notification
Notification Title	String	Title or subject of the notification
Notification Body	String	Main content or message of the notification
Notification Status	String	Status of the notification

Table 20. Data Dictionary for Announcement

Field Name	Data Type	Description
Id	Object id	A unique identifier for the delivery record
Title	String	Title of announcement

Body	string	Body of announcement
Author	String	Author of announcement

Table 21. Data Dictionary for office

Field Name	Data Type	Description
Id	Object id	A unique identifier for the product
Office Id	String	Unique identification for the office
Office Name	String	Name of the office to which approves the clearance
Office Info	String	Additional information about the office to which approves the clearance
Clearance Step	String []	The step of the clearance process

Table 22. Data Dictionary for clearance

Field Name	Data Type	Description
Id	Object id	A unique identifier for the transaction record
Reason	String	The reason for clearance
issueDate	String	The starting date of the clearance process
approvalDate	String	The date that you finished the clearance process
clearanceStatus	String	The status of the clearance process

Table 23. Data Dictionary for certificate

Field Name	Data Type	Description
Id	Object id	A unique identifier for the product record
Generate Date	String	The date that certificate is generated
Issuer Id	String	The user id that generates the certificate
Subject	String	The subject of the certificate
Stamp Id	String	The stamp that is put on certificate

Table 24. Data Dictionary for report

Field Name	Data Type	Description
Id	Object id	A unique identifier for the record
Report Type	String	The type of report
Report Body	String	Content of report body

Table 4- 1 Data Dictionary for Chat

Field Name	Data Type	Description
Id	Object id	A unique identifier for the chat
Date	Date	The date associated with the chat
Status	String	The status associated with the chat

Table 4- 2 Data Dictionary for help

Field Name	Data Type	Description
Id	Object Id	A unique identifier for the record
FAQ Id	String	Id for the frequently asked question
FAQ Content	String	Content for the frequently asked question

4.3. Dynamic model

4.3.1. Sequence diagram

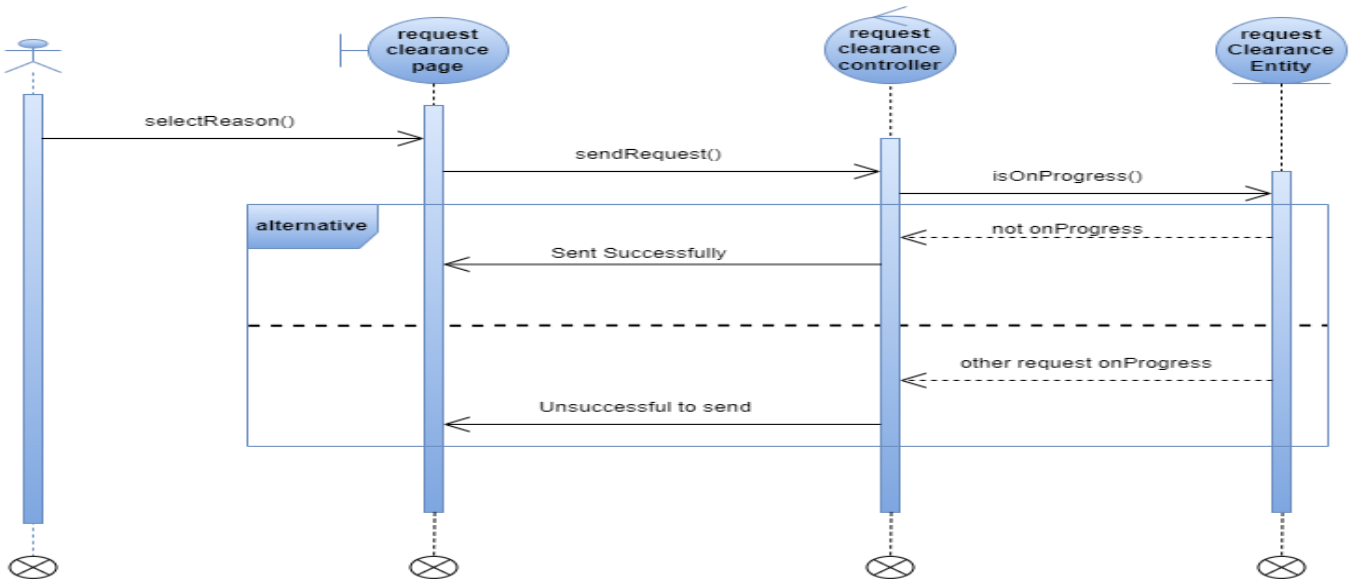


Figure 6. Request clearance sequence diagram

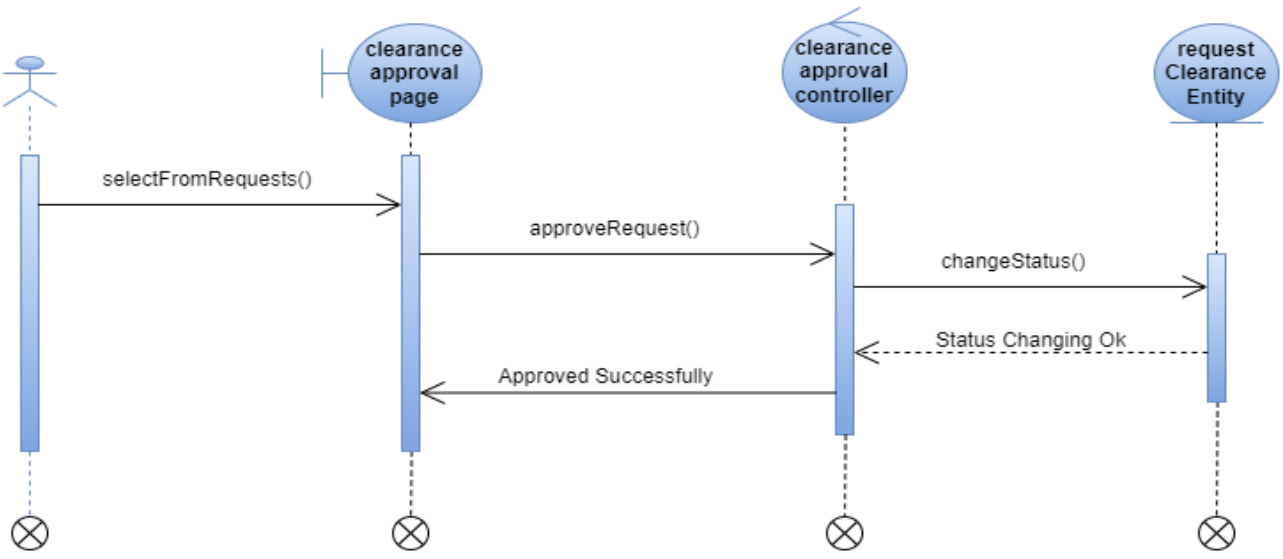


Figure 7. Approve request sequence diagram

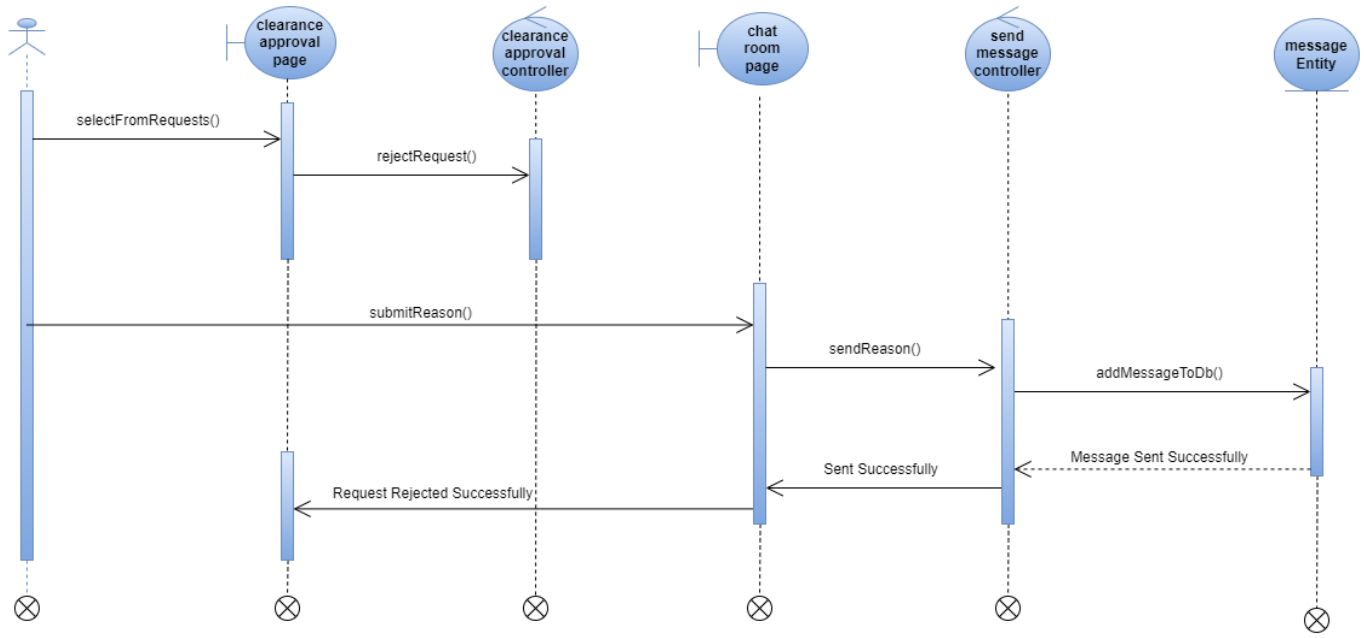


Figure 8. Reject request sequence diagram

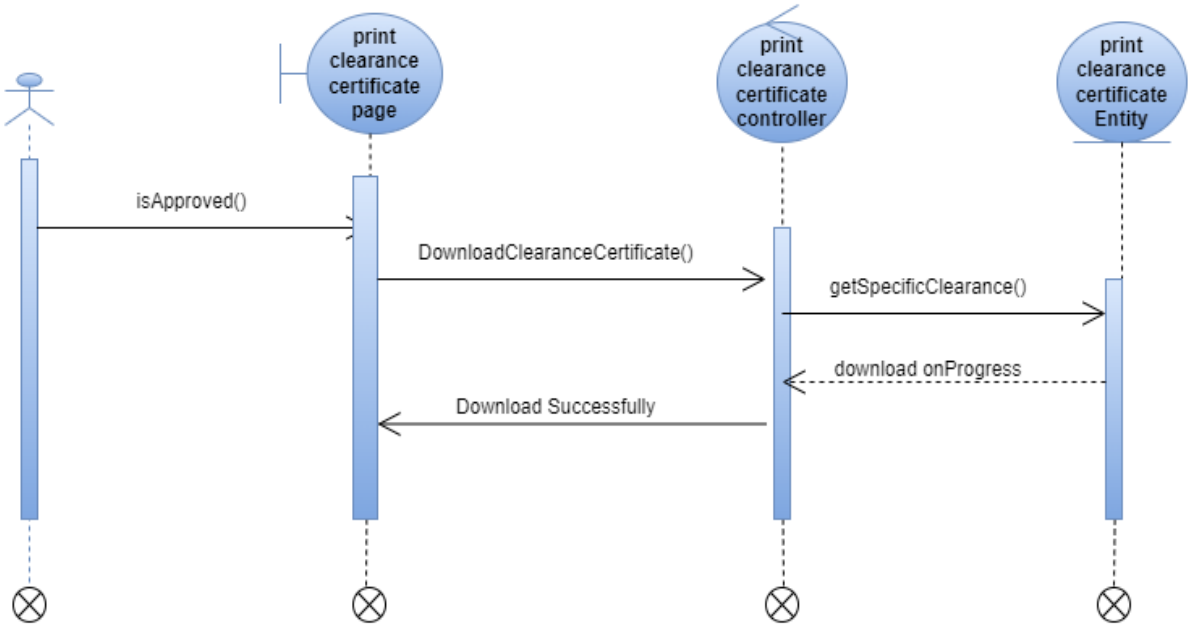


Figure 9. Print clearance certficate sequence diagram

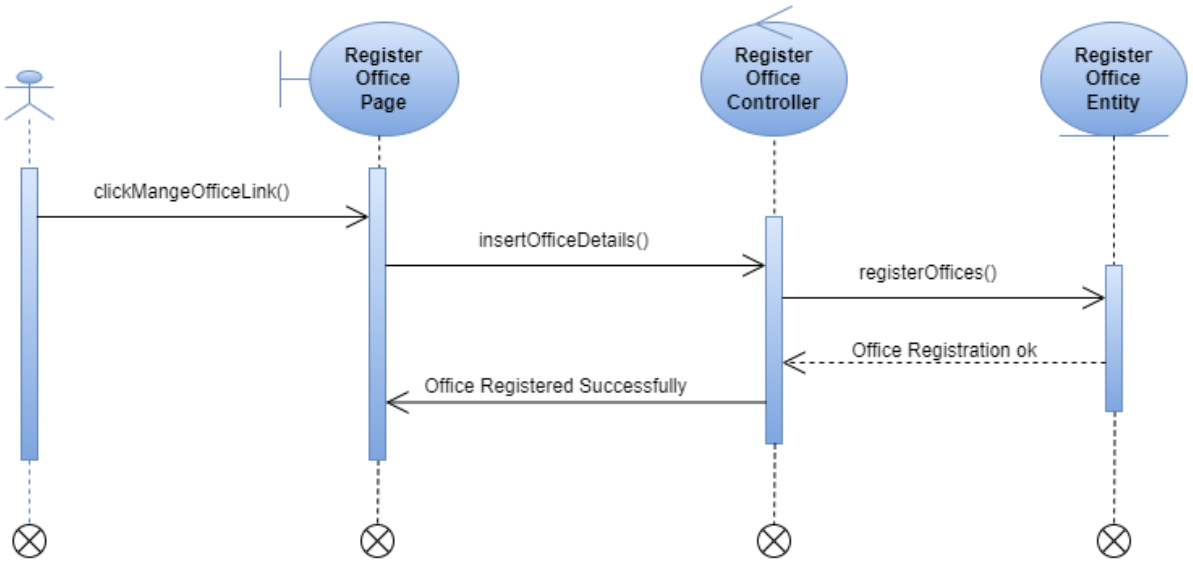


Figure 10. Register Office sequence diagram

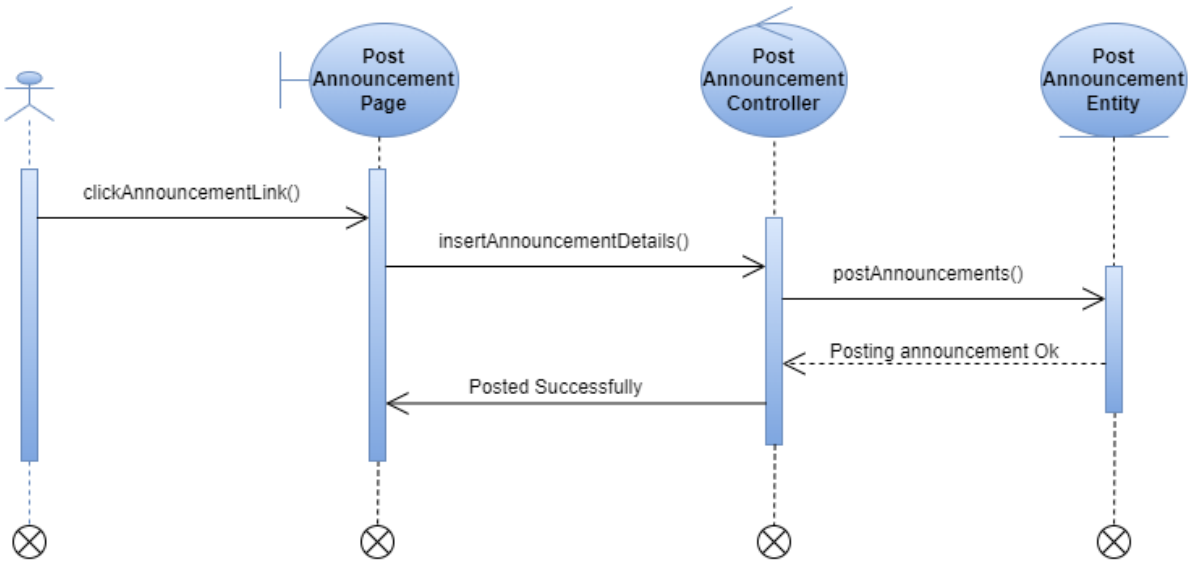


Figure 11. Post Announcements sequence diagram

4.3.2. Activity Diagram

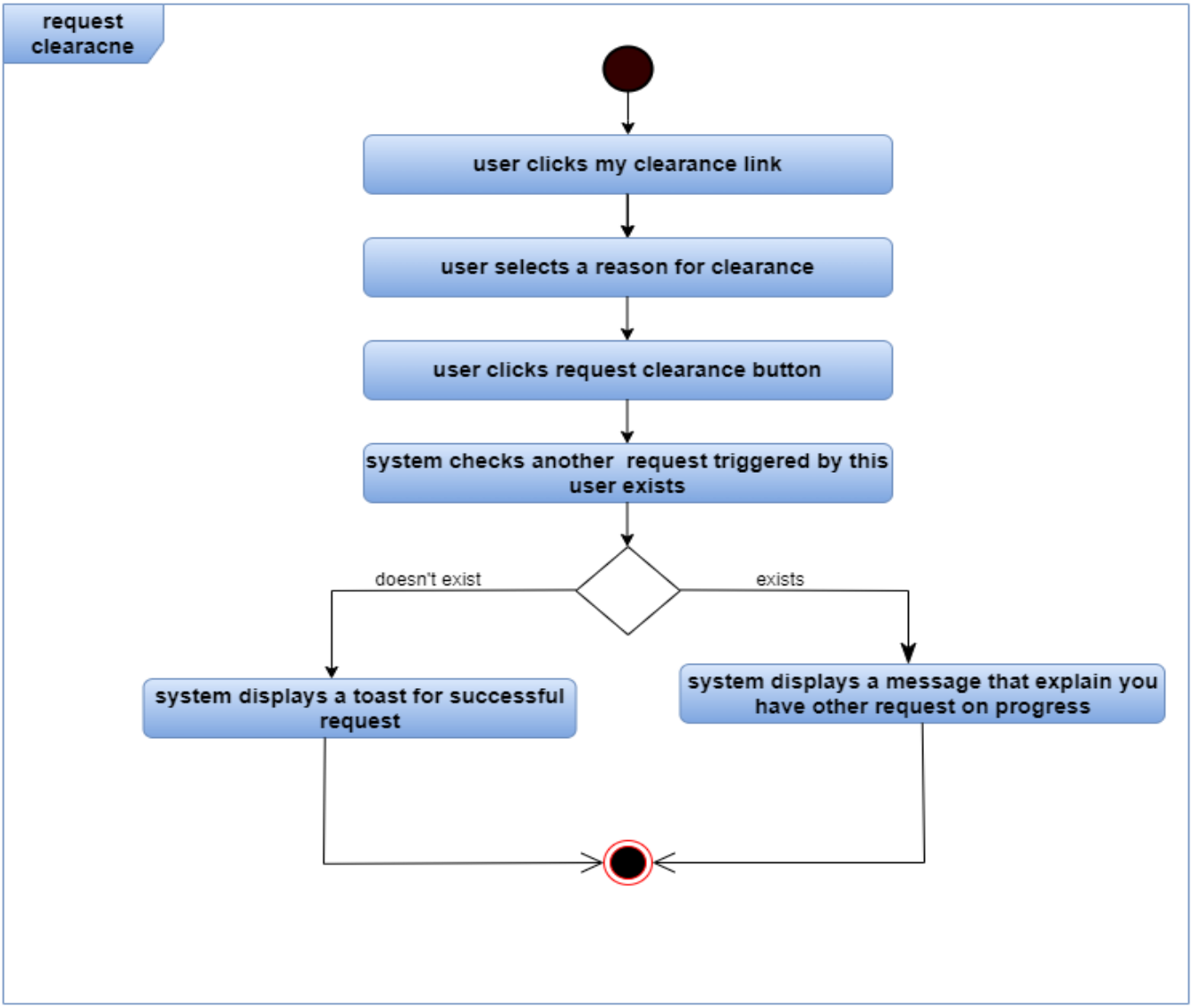


Figure 12. Request clearance activity diagram

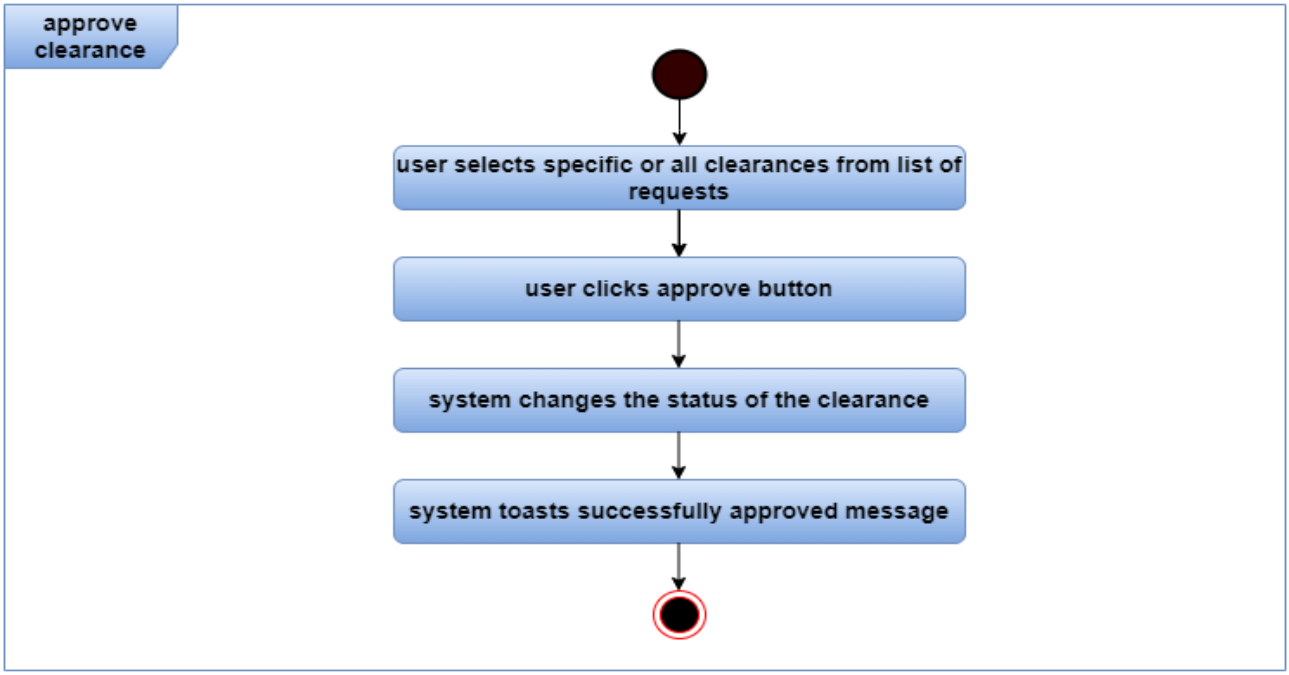


Figure 13. Approve clearance activity diagram

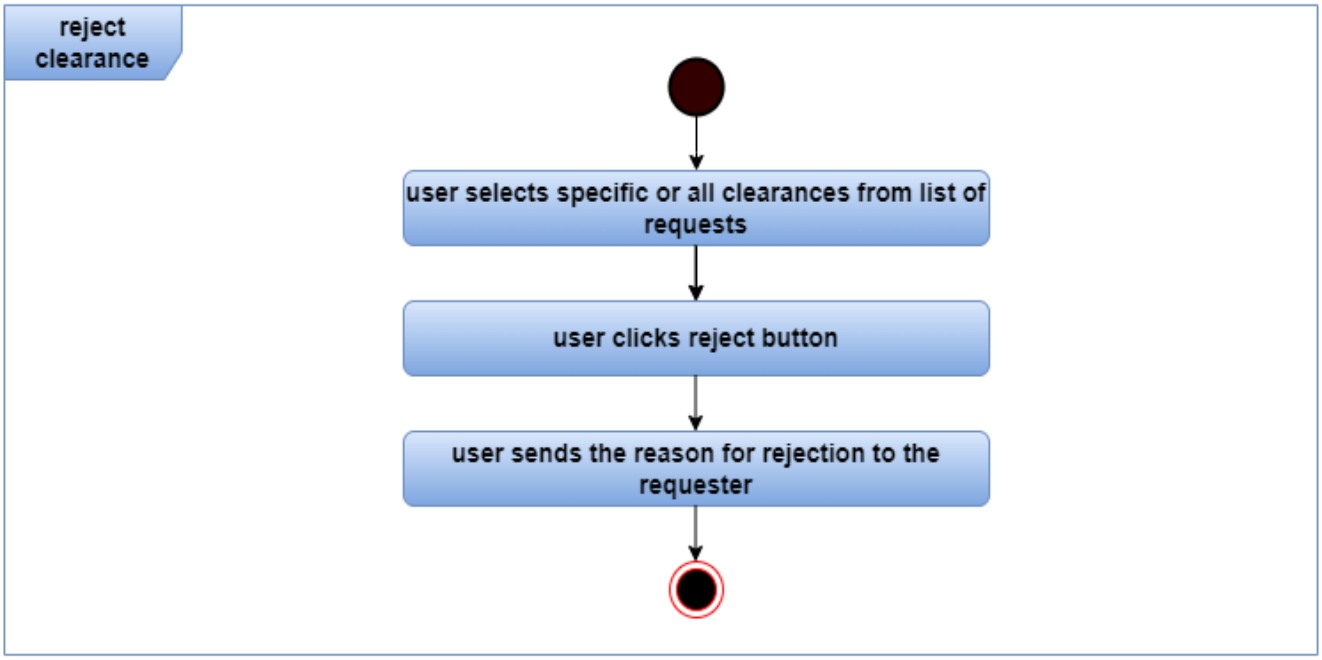


Figure 14. Reject clearance activity diagram

4.3.3. State diagram

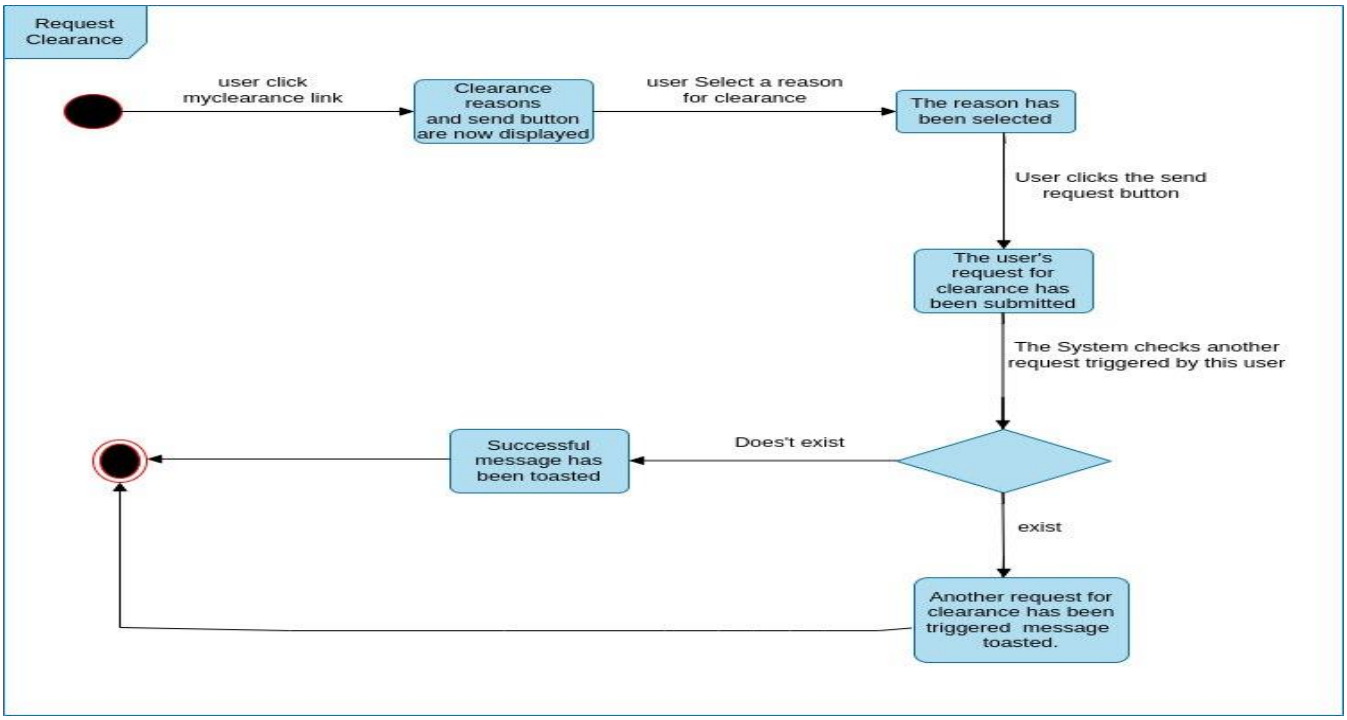


Figure 15. Request clearance state diagram

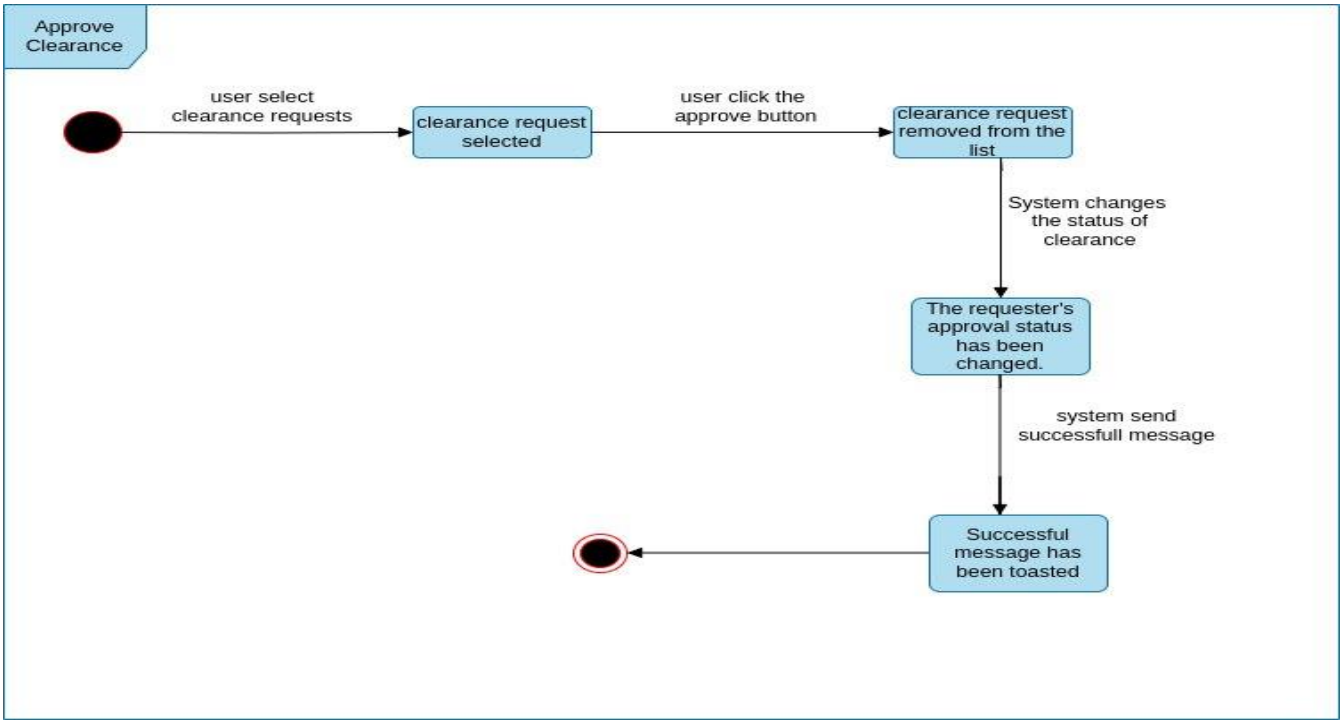


Figure 16. Approve clearance state diagram

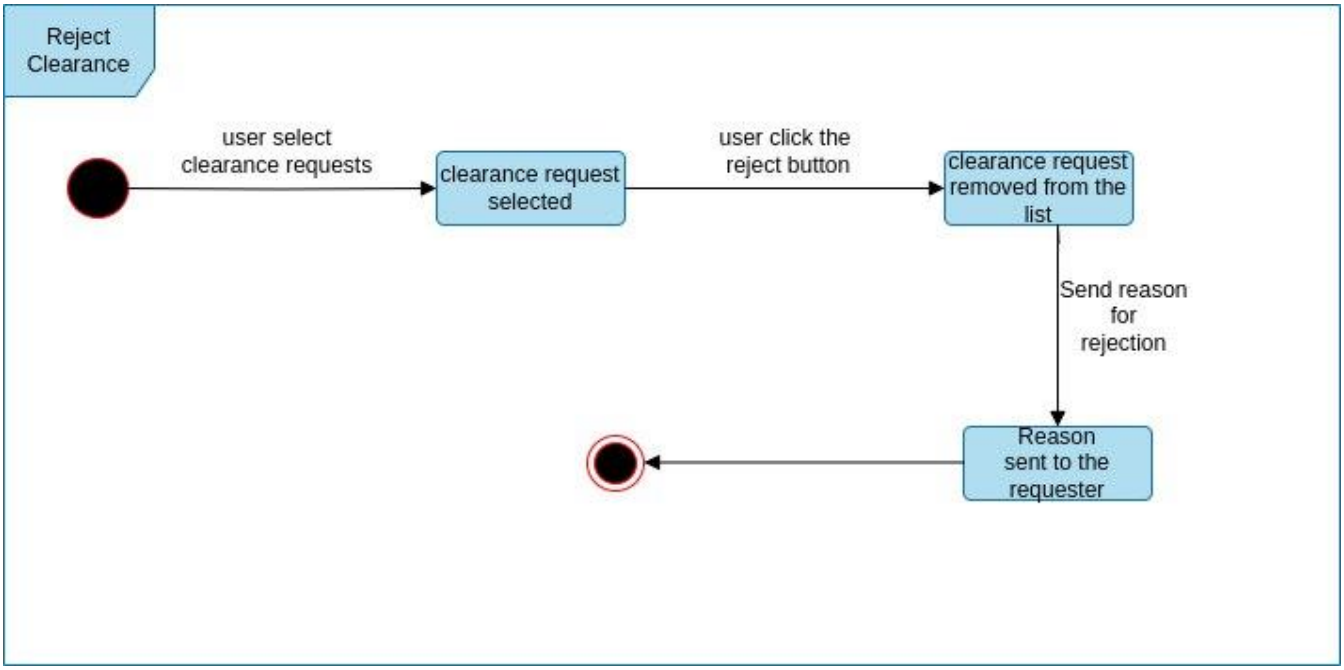


Figure 17. Reject clearance state diagram

Chapter 5

5. System Design

5.1. Design Goals

The Design Goals specify the qualities of the Web-based clearance management system that should be achieved and addressed during the design and implementation phase.

User Interface and Human Factors

To enable simple interactions with the system, the design will include an intuitive graphical user interface (GUI). Users will be guided through the process of entering their information by the system's interface, which will include distinct buttons that offer explicit direction on important tasks. The interface will also have intelligent colour selection, frames, and icons to improve aesthetic appeal. The interface will include an easy-to-use information flow that makes it evident to users how to navigate, guaranteeing a smooth and captivating user experience.

Security Issues

We are utilizing Next Auth to create authentication and authorization procedures in order to guarantee the security of our information system and satisfy the essential requirements of confidentiality, integrity, and availability. By assigning users a username and password, these systems ensure that only those who have been duly verified and granted permission can access the system. To ensure password security in the database, we utilize the bcrypt encryption method. Encrypting user passwords using this technique helps to provide a strong protection against possible attackers.

Performance Consideration

Next.js is a popular React framework that simplifies the process of building React applications. It provides a set of conventions and features that make it easier to create **performant**, **scalable**, and **SEO-friendly** web applications. One of the key advantages of Next.js is its ability to optimize response time through various mechanisms.Here's an overview of how Next.js can be used to improve response time in your project:

- **Server-Side Rendering (SSR):** Next.js supports server-side rendering, allowing your React components to be rendered on the server rather than in the browser. This can significantly reduce the time it takes for the initial page to load since the server can send a fully rendered HTML page to the client.
- **Automatic Code Splitting:** Next.js automatically splits your JavaScript bundles into smaller, optimized chunks. This means that only the necessary code for a particular page or component is sent to the client, reducing the initial load time.
- **Incremental Static Regeneration (ISR):** Next.js introduces Incremental Static Regeneration, a feature that allows you to statically generate parts of your site at build time and update them on-demand as traffic comes in. This helps in serving pre-rendered pages for static content while ensuring that dynamic content is always up-to-date.
- **Prefetching:** Next.js supports automatic prefetching of linked pages. When a user hovers over a link, Next.js can automatically fetch the required data for that page in the background, reducing the perceived loading time when the user actually clicks on the link.
- **API Routes:** Next.js enables you to create API routes that run server-side logic. This can be leveraged to offload certain operations to the server, reducing the workload on the client and improving response times for specific functionalities.
- **Custom Server Configuration:** If needed, Next.js allows you to customize the server configuration to optimize performance for your specific project requirements. This includes tweaking caching mechanisms, adjusting compression settings, and more.
- **Optimized Asset Handling:** Next.js optimizes the handling of assets like images and stylesheets. Images are automatically optimized, and stylesheets can be automatically configured for minimal impact on page load times.
- **Performance Analytics:** Next.js provides built-in performance analytics, helping you identify and address potential bottlenecks in your application. This can be crucial for optimizing response times.

- **CDN Support:** Next.js applications can be easily integrated with Content Delivery Networks (CDNs) for improved global response times. CDN caching helps in serving static assets from edge locations closer to the user.

Error handling and Validation

Next.js plays a crucial role in error handling and validation, providing developers with tools and conventions that contribute to a more robust and secure web application. One key aspect is its support for server-side rendering (SSR), which allows errors to be caught on the server before rendering a page, enhancing the overall stability of the application. Next.js also facilitates efficient client-side error handling through the use of React error boundaries, enabling developers to gracefully capture and manage errors in components. Additionally, Next.js supports custom error pages, allowing developers to create user-friendly error messages and maintain a consistent look and feel across the application. When it comes to validation, Next.js integrates seamlessly with popular validation libraries (YUP,Formik,JOI,Validator JS) and frameworks, enabling developers to implement robust client-side and server-side validation strategies. This ensures that data submitted by users is thoroughly checked for accuracy and adherence to predefined rules, minimizing the risk of security vulnerabilities and enhancing the overall reliability of the application.

Hardware Consideration

Scalability

We're designing with growth in mind, using a client-server architecture to scale horizontally. Regular checks will help us adjust resources as needed.

- **Horizontal Scalability:** MongoDB is designed for horizontal scalability, allowing you to distribute data across multiple servers or clusters. This means that as your data grows, you can add more servers to your MongoDB deployment to handle increased load, ensuring the database scales with the application.
- **Sharding:** MongoDB supports sharding, a method of distributing data across multiple machines. Sharding allows you to partition your data and distribute it across different nodes, enabling efficient read and write operations as the application grows.

Quality Considerations

MongoDB and Next.js can be leveraged in our project to enhance availability, reliability and User Operability.

Availability

- **MongoDB Replication**: Set up MongoDB replication to create multiple copies of your data across different servers or nodes. This ensures high availability by allowing the application to continue functioning even if one server goes down. MongoDB automatically promotes a secondary node to primary in case of a primary node failure.

Reliability

- **MongoDB Clustering and Sharding:** Implement MongoDB clustering and sharding to distribute your data across multiple servers or clusters. This not only improves performance but also enhances reliability by preventing a single point of failure. It ensures that your database can handle increasing amounts of data and traffic reliably.
- **Server-Side Rendering (SSR) in Next.js:** Implement SSR in Next.js to pre-render pages on the server, improving reliability by reducing the load on the client. Users receive fully rendered HTML pages, enhancing the consistency and reliability of the user experience.

User Operability

- **Responsive UI with Next.js:** Develop a responsive user interface using Next.js, allowing users to interact with the application seamlessly across various devices. Next.js supports the creation of dynamic and engaging user interfaces, contributing to user operability.
- **Real-Time Data Updates with MongoDB Change Streams:** Use MongoDB Change Streams to enable real-time updates in your application. This feature allows you to push changes in the database to the application in real time, enhancing user operability by providing instant updates without requiring manual refreshes.

Backup and Recovery

Implementing robust backup and recovery strategies is crucial for ensuring the integrity and availability of data in applications that use MongoDB and Next.js. Some technologies are:-

- ✓ **MongoDB Atlas Backup:**
  - ➢ **Automated Backups**: MongoDB Atlas, a fully managed cloud database service, provides automated backups. It regularly takes snapshots of your data and allows you to restore to specific points in time, providing a convenient and reliable backup solution.
  - ➢ **Point-in-Time Recovery:** MongoDB Atlas supports point-in-time recovery, enabling you to restore your data to a specific moment, minimizing potential data loss in the event of issues or accidental data corruption.
- ✓ **Database Dump and Restore:**
  - ➢ **mongodump and mongorestore:** MongoDB includes command-line tools like mongodump and mongorestore that allow you to create and restore BSON data dumps. This method is useful for creating manual backups outside of automated solutions.

5.2. Proposed System Architecture

- Designing the architecture for WKUCMS, the proposed system architecture establishes a robust framework for the web application. Embracing a 3-tier architecture, this design separates presentation, application, and data layers. The modular structure enhances scalability and code reusability, crucial for WKUCMS's dynamic content and evolving features. With this approach, the system achieves a balance between flexibility and maintenance efficiency, ensuring seamless adaptability to future enhancements and changes in requirements. The proposed architecture aligns with the project's goals, providing a solid foundation for the development of WKUCMS. We choose three-tier architectures for the following reasons:

- ➢ **Modularity and Separation of Concerns:**
  - The three-tier architecture separates the system into distinct layers: Presentation, Application, and Data. This modularity allows for the independent development, testing, and maintenance of each tier, fostering a clear separation of concerns.

- **Scalability:**
    - The three-tier model supports scalability by enabling the independent scaling of each layer based on specific needs. For instance, if the user base increases, the Presentation Tier (front end) can be scaled independently of the Application Tier (business logic) and the Data Tier (database).
- **Ease of Maintenance and Updates:**
    - Updates and maintenance tasks become more manageable in a three-tier architecture. Changes to the user interface (Presentation Tier) don't necessarily impact the underlying business logic (Application Tier) or the data storage (Data Tier). This separation simplifies the maintenance process.
- **Flexibility and Adaptability:**
    - The modular nature of three-tier architecture makes it adaptable to changing requirements. You can update or replace one tier without affecting the others, allowing for more flexibility in incorporating new features or technologies.
- **Improved Security:**
    - Security measures can be applied more effectively at each tier. For example, access controls and authentication mechanisms can be implemented in the Presentation Tier, while encryption and data validation can be enforced in the Data Tier. This layered approach enhances overall system security.
- **Enhanced Development Workflow:**
    - The three-tier model promotes a collaborative development workflow. Front-end developers can work on the user interface, back-end developers can focus on business logic, and database administrators can manage the data layer. This specialization improves efficiency and reduces bottlenecks.
- **Interoperability and Integration:**
    - The modular design facilitates the integration of third-party services and external systems. The Application Tier serves as an intermediary for communication

between the Presentation Tier and Data Tier, allowing seamless integration with other systems.

Let us see how we will use the 3 tier architecture in our project in detail like this:

Presentation Tier

The Presentation Tier, also known as the front end or user interface layer, is the part of a software application responsible for presenting information to the user and handling user interactions. In the context of our application using Next.js, the Presentation Tier is built using Next.js as the primary technology.

Application Tier

The Application Tier, also known as the business logic or backend layer, plays a crucial role in processing data, managing application functionality, and interacting with external resources. In the context of your application using Next.js, the Application Tier is built using serverless functions and API routes provided by Next.js, connecting the frontend (Presentation Tier) to the backend services.

Data Access Tier

In a Next.js application where MongoDB is used as the database, the Data Access Tier plays a crucial role in managing data interactions, handling database operations, and ensuring seamless communication between the frontend (Presentation Tier) and the MongoDB database.

Utilize serverless functions or API routes within Next.js as the foundation of the Data Access Tier. These functions are responsible for handling data access, processing requests from the frontend, and interacting with the MongoDB database.

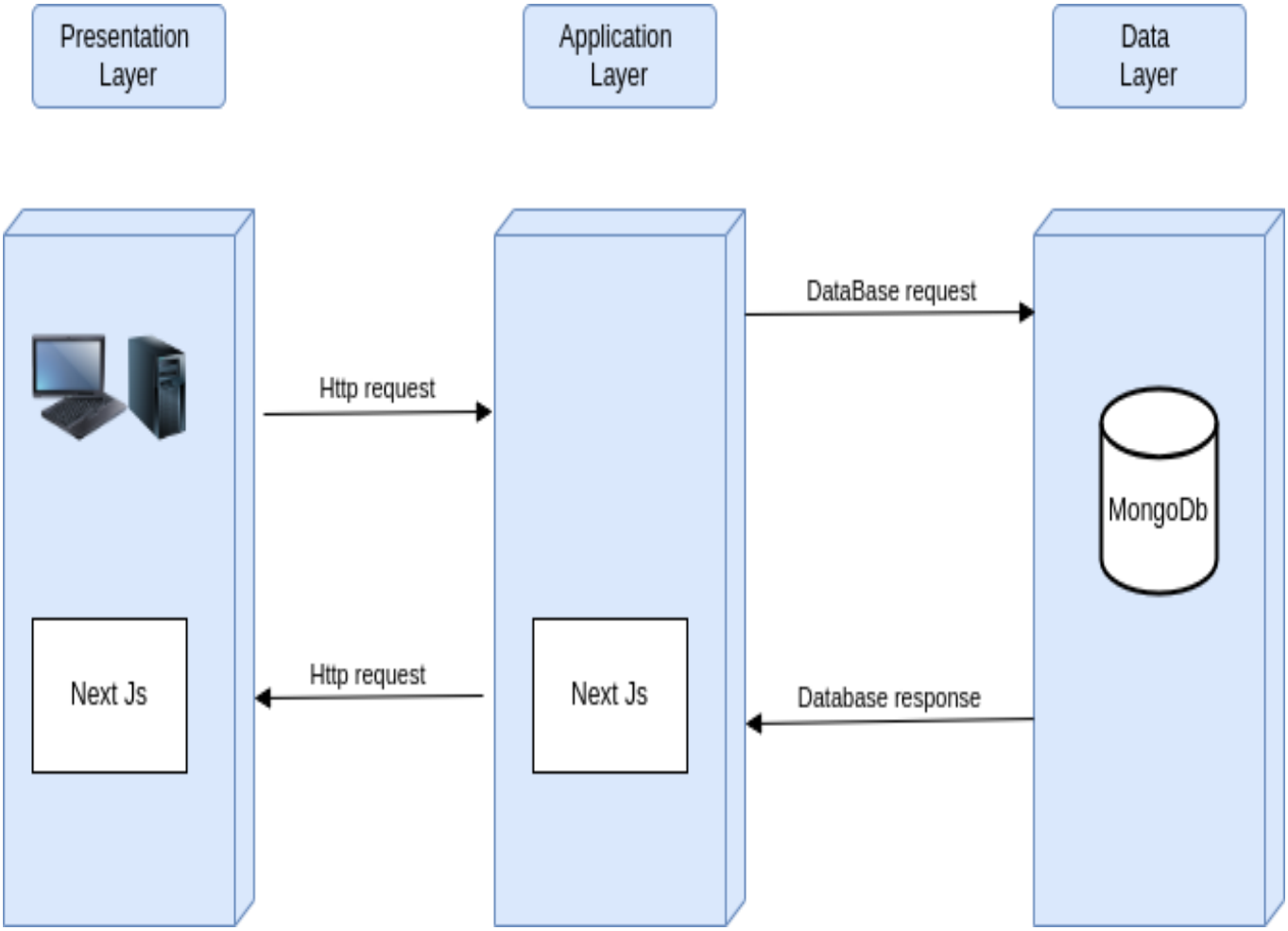


Figure 18. Proposed System Architecture

5.2.1. Subsystem Decomposition and Description

1. User Management Subsystem

- Description: Handles user registration, and profile management for users.
- Components:
  - User Registration Module
  - User Data

2. Clearance Subsystem

- Description: manages approval clearance, rejection clearance, request clearance, clearance status, clearance history

- Components:
    - Approval clearance
    - Rejection clearance
    - Request clearance
    - Clearance status
    - Clearance history

3. Certificate Subsystem

- **Description**: This subsystem is responsible for creating and printing certificates for various purposes, such as graduation,scholarship and so on.
- **Components**:
    - Generate certificate
    - Print certificate

4. Feedback and announcement

- **Description**: This subsystem is responsible for Sending announcements and Sending feedbacks for various purposes.
- **Components**:
    - Send announcement
    - Send feedback

5. Communication

- **Description**: This subsystem is responsible for Sending messages and Sending Notifications for various purposes.
- **Components**:
    - Send Message
    - Send Notification

6. Security

- **Description**: This subsystem is responsible for authentication and authorization and data Encryption.
- **Components**:
  - Authentication and authorization
  - Data Encryption

7. System administration

- **Description**: This subsystem is responsible for maintenance, updates and System configuration.
- **Components**:
  - Maintenance and Updates
  - System configuration

8. Generate Report

- **Description**: Generates reports and analytics.
- **Components**:
  - Data Visualizations
  - Generate Report

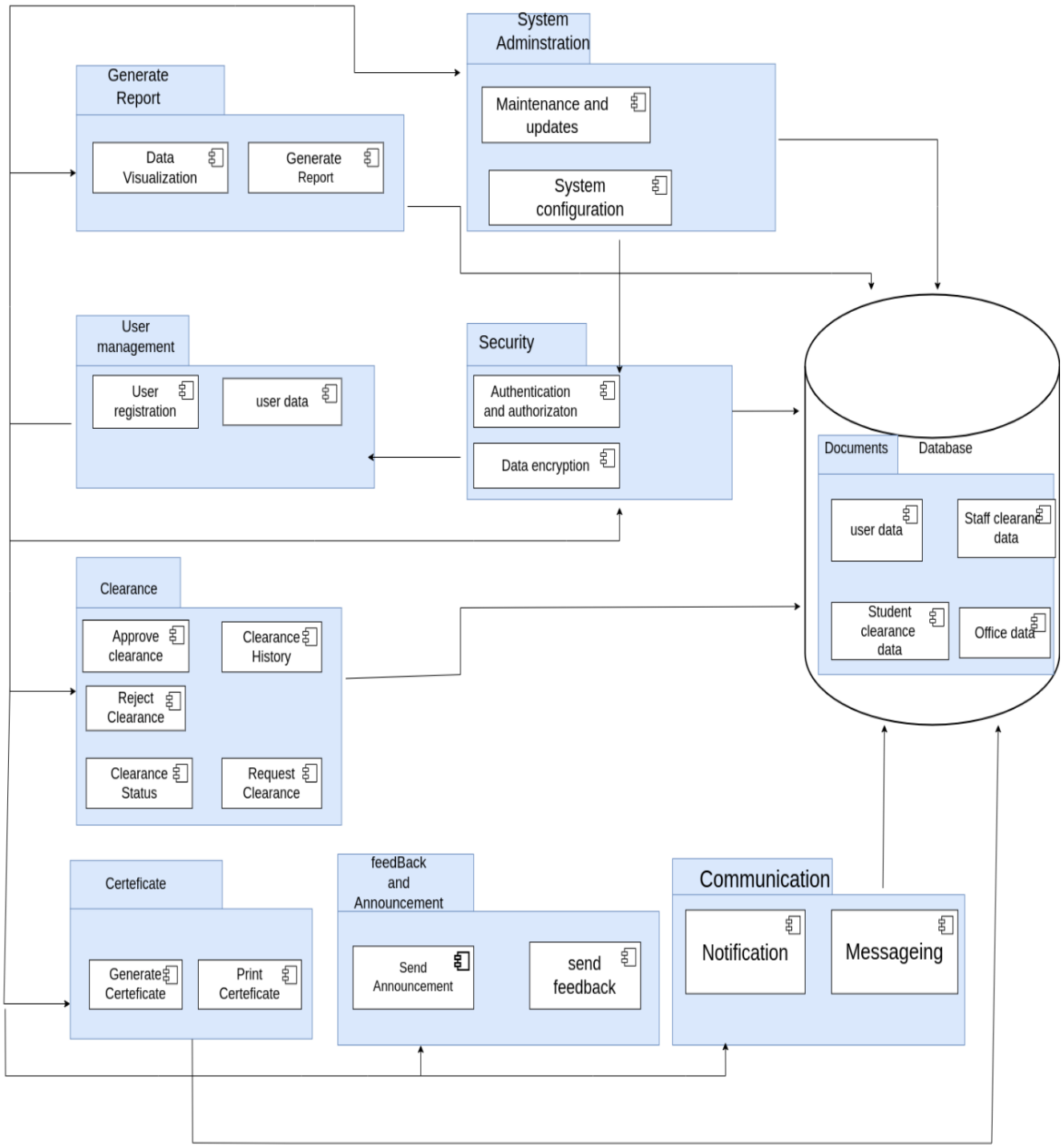


Figure 19. Component Diagram

5.2.2. Hardware/Software Mapping

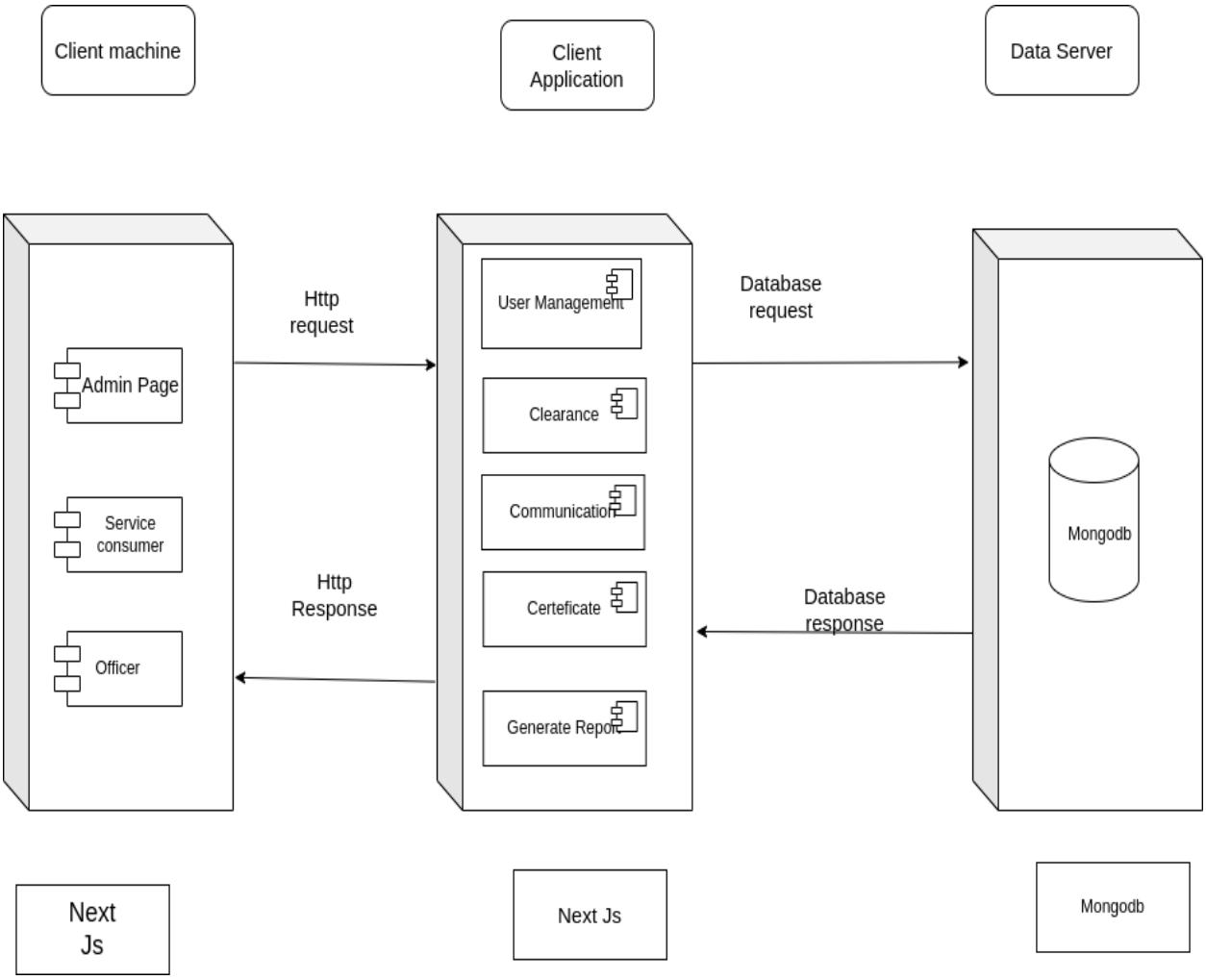


Figure 20. Hardware /Software mapping

5.2.3 Detailed Class Diagram

Class diagrams for NoSQL-based systems may differ from those for traditional SQL-based systems due to the fundamental differences in their data models and structures.

In SQL databases, data is typically organized into tables with predefined schemas that enforce relationships and constraints. Class diagrams in SQL-based systems often represent these tables, their columns, relationships between tables (via foreign keys), and other constraints.

However, in NoSQL databases, there isn't a fixed schema, and data is often stored in a schema-less or schema-flexible manner, such as in key-value, document, column-family, or graph-based models. Class diagrams for NoSQL systems might represent entities, their attributes, and associations, focusing more on the logical or conceptual structure rather than strict table relationships. Here is the detailed class diagram that is designed for wkucms:

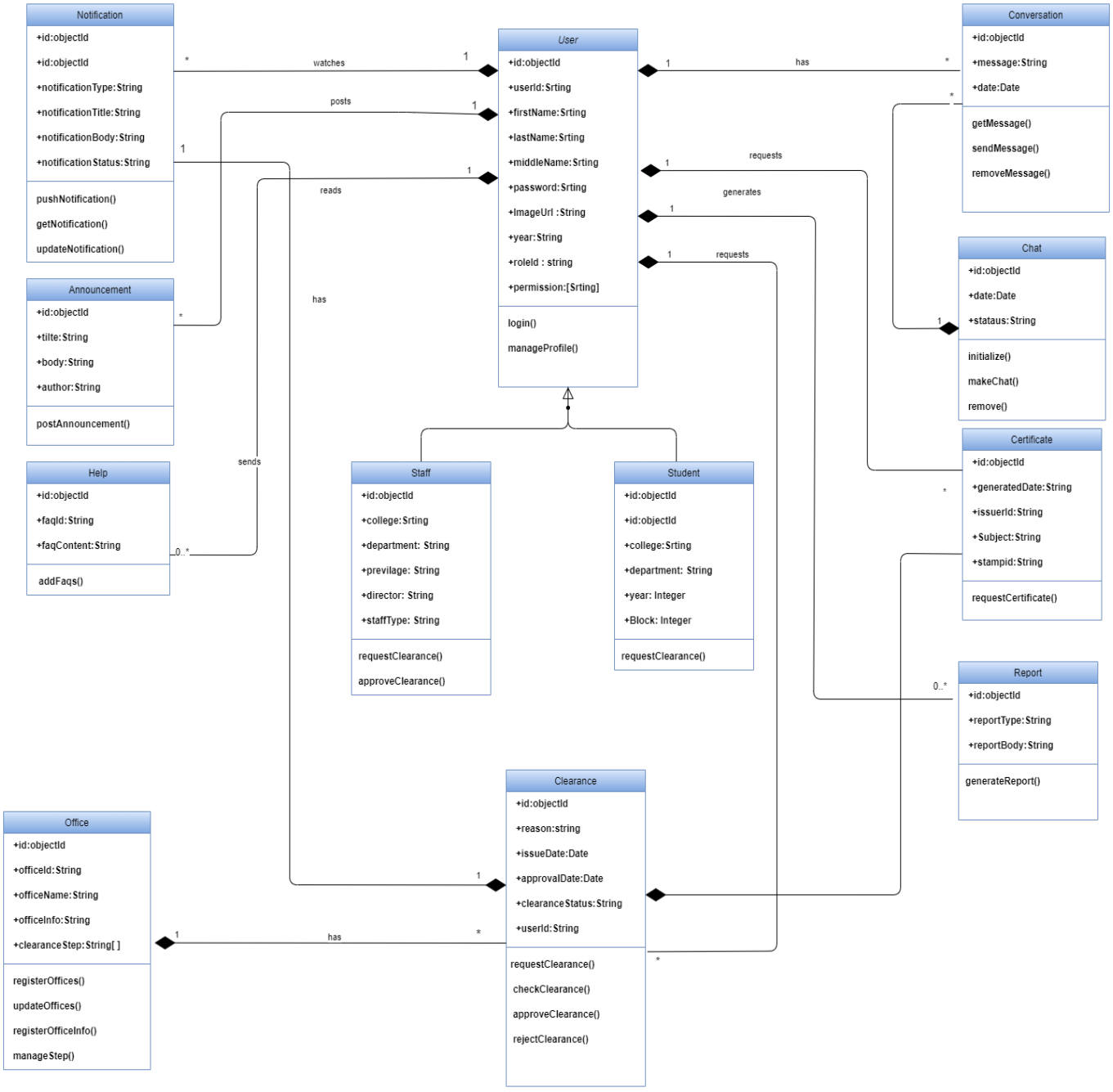


Figure 21. Detailed class diagram

5.2.4 Persistent Data Management

Persistent data management involves utilizing MongoDB as a NoSQL database to store and retrieve data persistently. It provides a basic overview of how you might organize different schemas within MongoDB using separate collections. MongoDB is known for its flexibility, scalability, and ability to handle diverse data types. Here is our persistent diagram for wkucms:

Figure 22. Persistent data management

5.2.5. Access control and Security

Access control and security are paramount considerations in a clearance management system to ensure that sensitive information is protected, and only authorized personnel have access to specific resources. Here's an overview of access control and security measures within a clearance management system:

Access Control:

Role-Based Access Control (RBAC):

Implement RBAC to assign specific roles and permissions to users based on their responsibilities and clearance levels. Different roles may have varying levels of access to system functionalities and sensitive data.

Access Approval Workflows:

Implement approval workflows for granting and modifying access rights. Clearance changes or access requests should be subject to approval by appropriate authorities to maintain security and compliance with organizational policies.

Table 25. Access Control Privileges

Actors	Access Controls
Student or staff	LogIn in to the system Request clearance View Clearance history View Clearance status View FAQ and Office description View announcements Print Clearance LogOut from the system

Officer	LogIn in to the system Approve the clearance request Reject the clearance request View requesters Request clearance View Clearance history View FAQ and Office description View announcements LogOut from the system
Admin	LogIn in to the system manage user LogOut from the system

5.3. Packages

This section breaks down subsystems into packages as well as expansion on code files organization. It gives a complete picture of each package and lists its dependencies with other packages as well as stating how it should be used. Using UML package diagrams, the document seeks to diagrammatically represent the structure of packages.

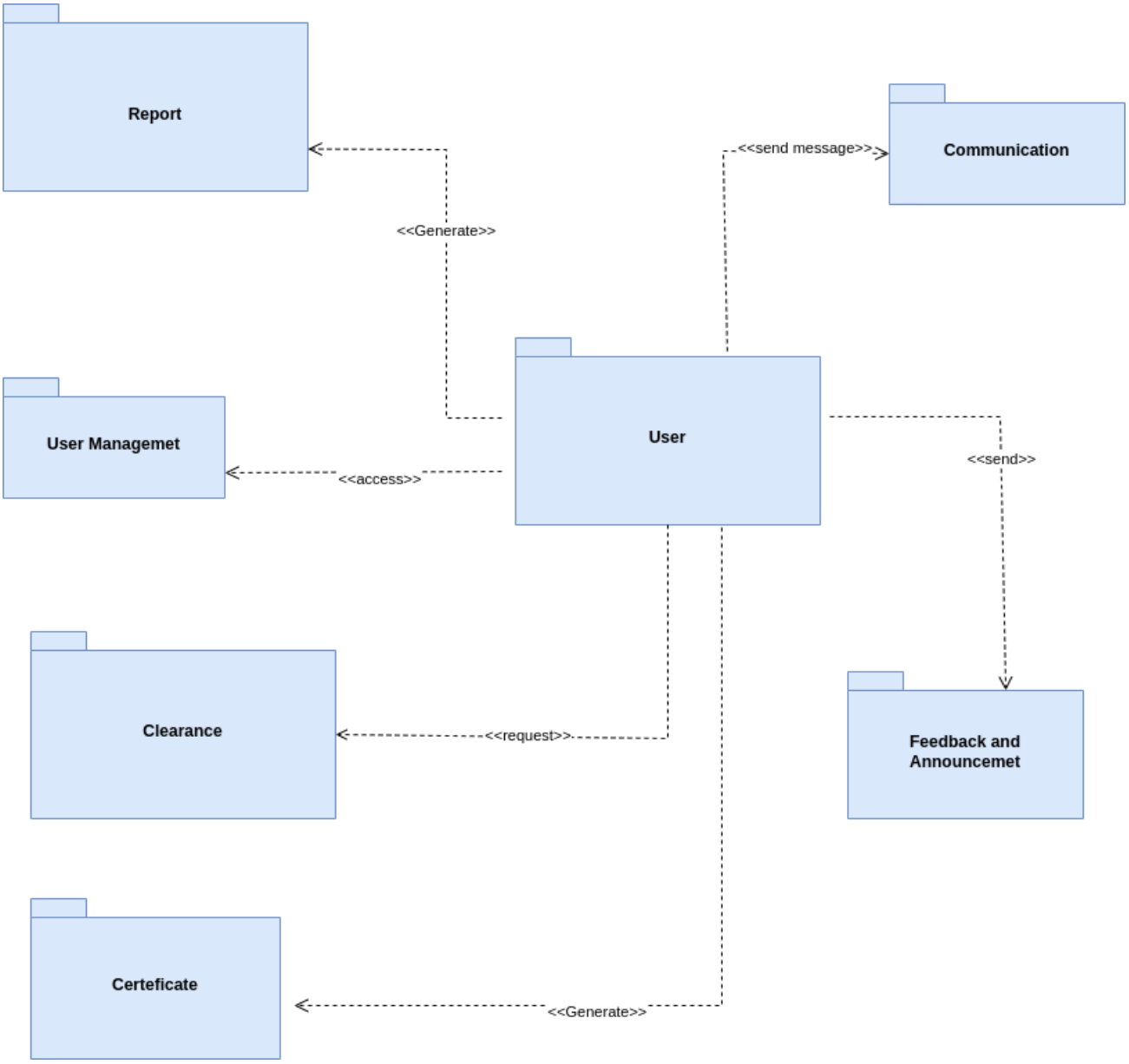


Figure 23. Package Diagram

5.4. Algorithm Design

❖ Pseudocode for requesting clearance

```
FUNCTION TaskItem():
  selectedTask = useState(null)
  session = useSession()
  userId = session?.data?.user.userId
  firstname = session?.data?.user.firstname
```

```
  middlename = session?.data?.user.middlename
  status = IF session?.data?.user.role == "STUDENT" THEN "HEAD" ELSE IF
session?.data?.user.role == "STAFF" THEN "HR" END IF

  FUNCTION handleTaskSelection(task):
    SET selectedTask to task

  FUNCTION handleSend():
    IF selectedTask != null THEN
      IF session?.data?.user.role == "STUDENT" THEN
        TRY:
          response = FETCH("/api/studentRequest", {
            method: "POST",
            body: JSON.stringify({
              userId: userId,
              reason: selectedTask,
              status: status,
              firstname: firstname,
              middlename: middlename,
              role: "STUDENT",
            }),
            headers: {
              "Content-Type": "application/json",
            },
          })

          IF response.ok THEN
            responseData = AWAIT response.text()
            TOAST.success(responseData)
          END IF
        CATCH error:
          TOAST.error("Invalid request")
          LOG(error)
        END TRY
      END IF

      IF session?.data?.user.role == "STAFF" THEN
        TRY:
          response = FETCH("/api/staffRequest", {
            method: "POST",
            body: JSON.stringify({
              userId: userId,
```

```
                    reason: selectedTask,
                    status: status,
                    firstname: firstname,
                    middlename: middlename,
                    role: "STAFF",
                }),
                headers: {
                    "Content-Type": "application/json",
                },
            })

            IF response.ok THEN
                responseData = AWAIT response.text()
                TOAST.success(responseData)
            END IF
        CATCH error:
            TOAST.error("Invalid request")
            LOG(error)
        END TRY
    END IF
ELSE
    TOAST.info("Select your reason first")
END IF

RETURN (
    <div className="md:mt-7 mt-4 md:py-7 py-4 2xl:h-[60vh] border shadow-default flex flex-col rounded-lg bg-white border-bodydark1 dark:border-strokedark dark:bg-boxdark">
        <div className="px-4 sm:px-7">
            <h5 className="dark:text-white font-satoshi text-4xl font-bold mb-4 text-primary sm:text-left">
                Choose the cause for your clearance
            </h5>
            <div className="pt-5 grid grid-cols-6 sm:gap-6 gap-4">
                <div className="md:ml-26 ml-4 grid grid-cols-1 lg:grid-cols-2 col-span-3 xl:gap-8 gap-5">
                    FOREACH req IN request:
                        <label htmlFor={req.value} className="cursor-pointer text-primary text-lg font-md" key={req.value}>
                            <div className="flex items-center pt-0.5">
                                <input
                                    type="radio"
                                    id={req.value}
```

```
                            name="task-radio-group"
                            value={req.value}
                            checked={selectedTask === req.value}
                            onChange={() => handleTaskSelection(req.value)}
                            className="dark:text-white marker:text-white"
                        />
                        <div className="dark:text-white box flex h-5 w-5 items-center justify-
center dark-border-strokedark">
                            <span
                                className={`text-white ${
                                    selectedTask === req.value ? "opacity-100" : "opacity-0"
                                }`}
                            ></span>
                        </div>
                        <p className="dark:text-white">{req.label}</p>
                    </div>
                </label>
                END FOREACH
            </div>
        </div>
        <button
            className="md:ml-26 ml-4 mt-10 flex text-lg justify-center rounded-lg bg-primary
py-3 px-6 font-semibold text-gray hover:bg-opacity-95"
            onClick={handleSend}
            type="submit"
        >
            Send Request
        </button>
    </div>
  </div>
 )
END FUNCTION

EXPORT DEFAULT TaskItem
```

- Pseudocode for requesting clearance

```
// Import necessary modules and components
USE CLIENT
```

```
IMPORT Filter FROM "@/components/Admin/Filter"
IMPORT UserContainer FROM "@/components/User/UserContainer/UserContainer"
IMPORT { Metadata } FROM "next"
IMPORT RegisterStudent FROM "@/components/Modals/RegisterStudent"
IMPORT useSWR FROM 'swr'
IMPORT { useState } FROM "react"

// Define office and college data
CONST officeData = [
  { id: 1, name: "Office 1" },
  { id: 2, name: "Office 2" },
  { id: 3, name: "Office 3" },
  // Add more office data here
]
CONST collegeData = [
  { id: 1, name: "College of Computing and Informatics" },
  { id: 2, name: "Engineering" },
  { id: 3, name: "Social sciences and Humanities" },
  { id: 4, name: "College of behavioral science" },
  // Add more college data here
]

// Define columns for the data table
CONST columns = [
  { field: "userId", headerName: "ID", width: "100" },
  { field: "firstname", headerName: "First name", width: "160" },
  { field: "middlename", headerName: "Last name", width: "160" },
  { field: "reason", headerName: "Reason", width: "160" },
  { field: "status", headerName: "Status", width: "160" },
]

// Define sample rows for the data table
CONST rows = [
  // Sample data rows
]

// Define a fetcher function for useSWR
FUNCTION fetcher(url):
RESPONSE = FETCH(url)
DATA = AWAIT RESPONSE.JSON()
UPDATED_DATA = MAP DATA TO USER => ({ ...USER, id: USER._id, roleId: USER._id
})
  RETURN UPDATED_DATA
```

```
// Define the ApproveStudent component
FUNCTION ApproveStudent():
// Use state for search term
[searchTerm, setSearchTerm] = USESTATE('')

// Use SWR to fetch and cache data with automatic
{ data: userData, error } = USESWR('http://localhost:3000/api/studentApproval', fetcher, {
  initialData: rows,
  revalidateOnFocus: false,
  refreshInterval: 2000, // Set the refresh interval in milliseconds (e.g., 10000 for 10 seconds)
})

// Log user data
LOG("session from approval ad ", userData)

  // Handle loading and fetch errors
  IF NOT userData AND NOT error THEN
RETURN < p > Loading...</ >
  END IF

  IF error THEN
LOG('Error fetching data:', error)
RETURN < p > Failed to fetch data</ >
  END IF

// Render the component
RETURN(
  <div className="bg-white sm:px-14 dark:bg-black dark:border-black">
    <h1 className="pt-8 pb-5 pl-4 font-extrabold text-4xl text-primary dark:text-
white">Student Approval</h1>
    <div className="pt-2 px-4 grid grid-cols-12 gap-4 md:gap-6 2xl:gap-4.5">
      <Filter officeData={officeData} collegeData={collegeData} />
      <UserContainer columns={columns} rows={userData} modal={RegisterStudent} />
    </div>
  </div>
)
END FUNCTION

EXPORT DEFAULT ApproveStudent
```

5.5. Interface Design

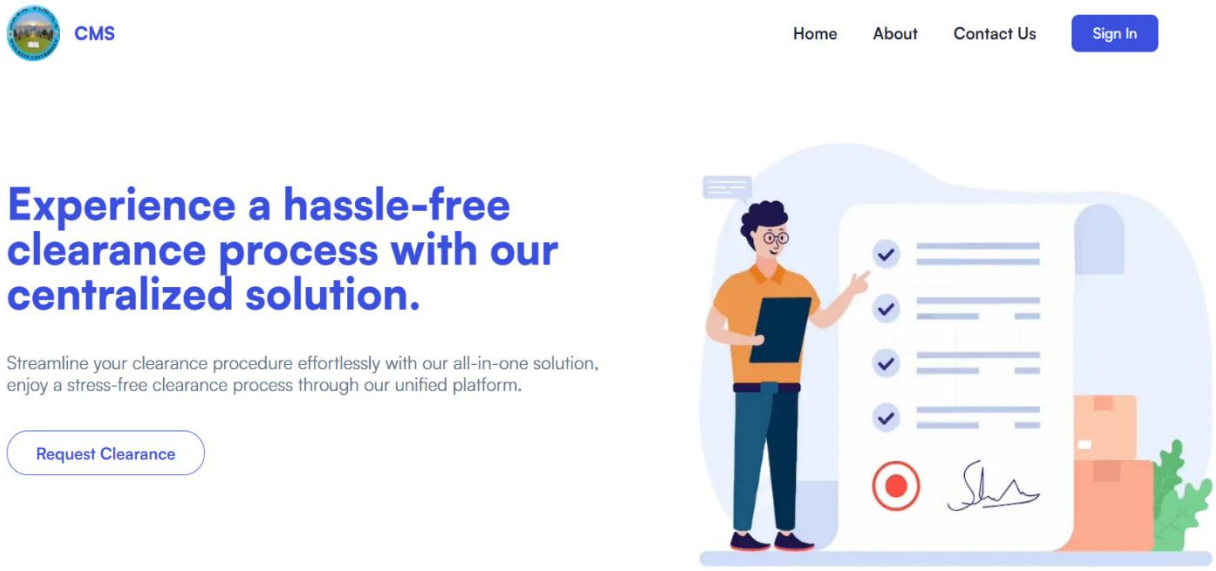


Figure 24. Landing page of wkucms

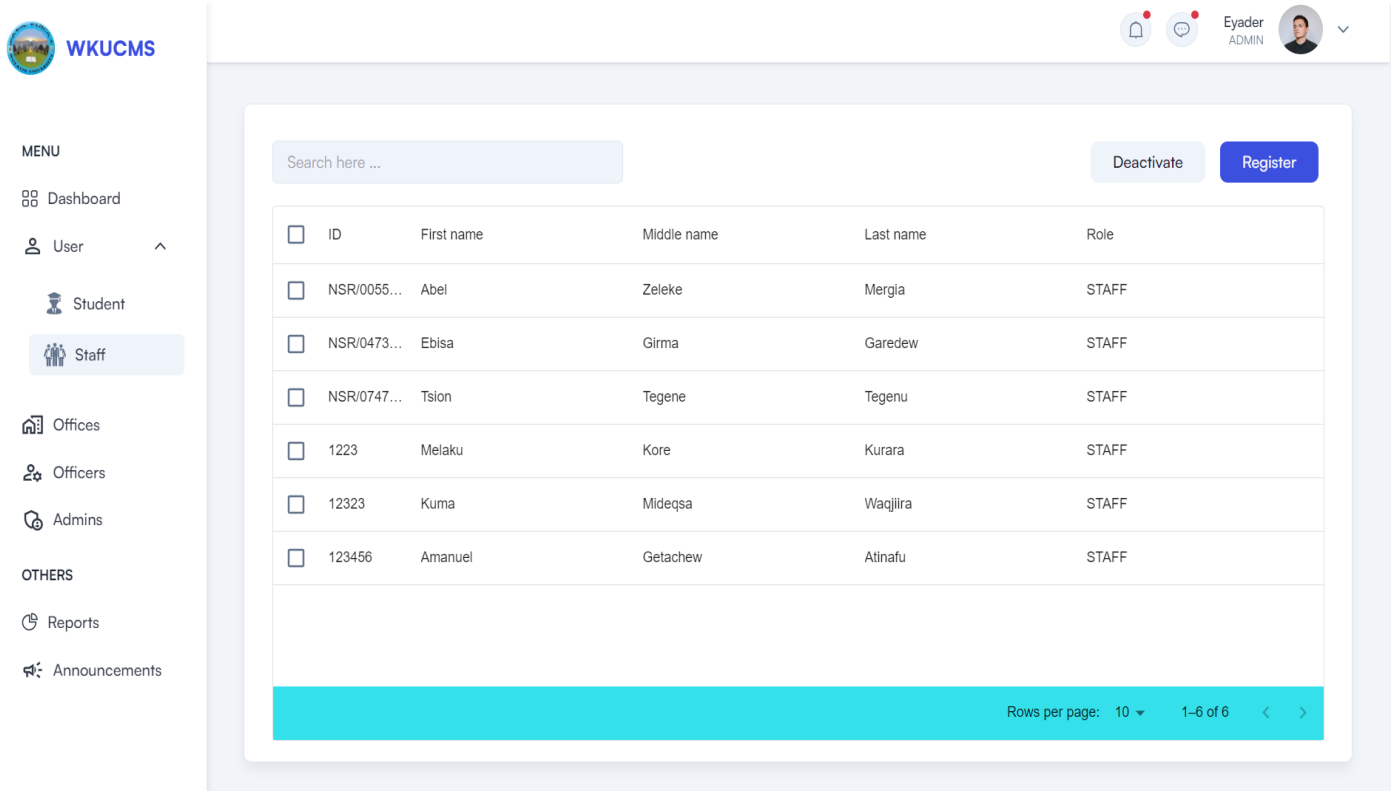


Figure 25. Admin side of wkucms

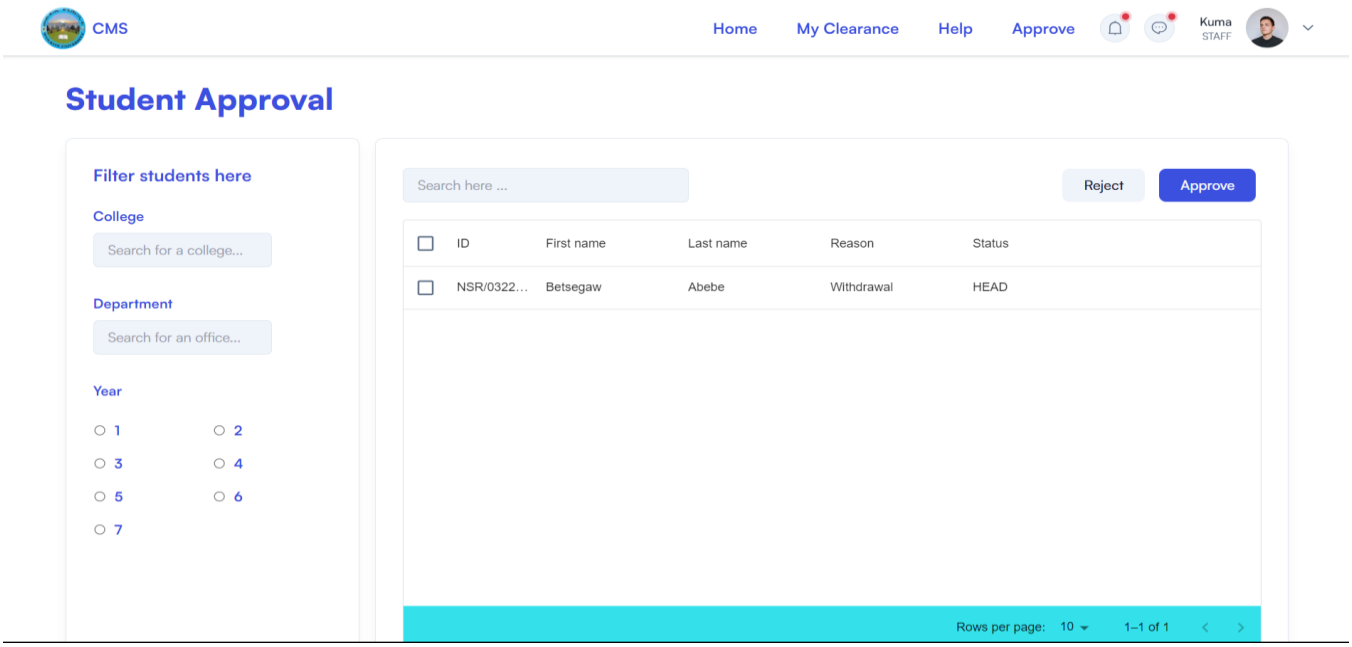


Figure 26. Approval/rejection page of wkucms

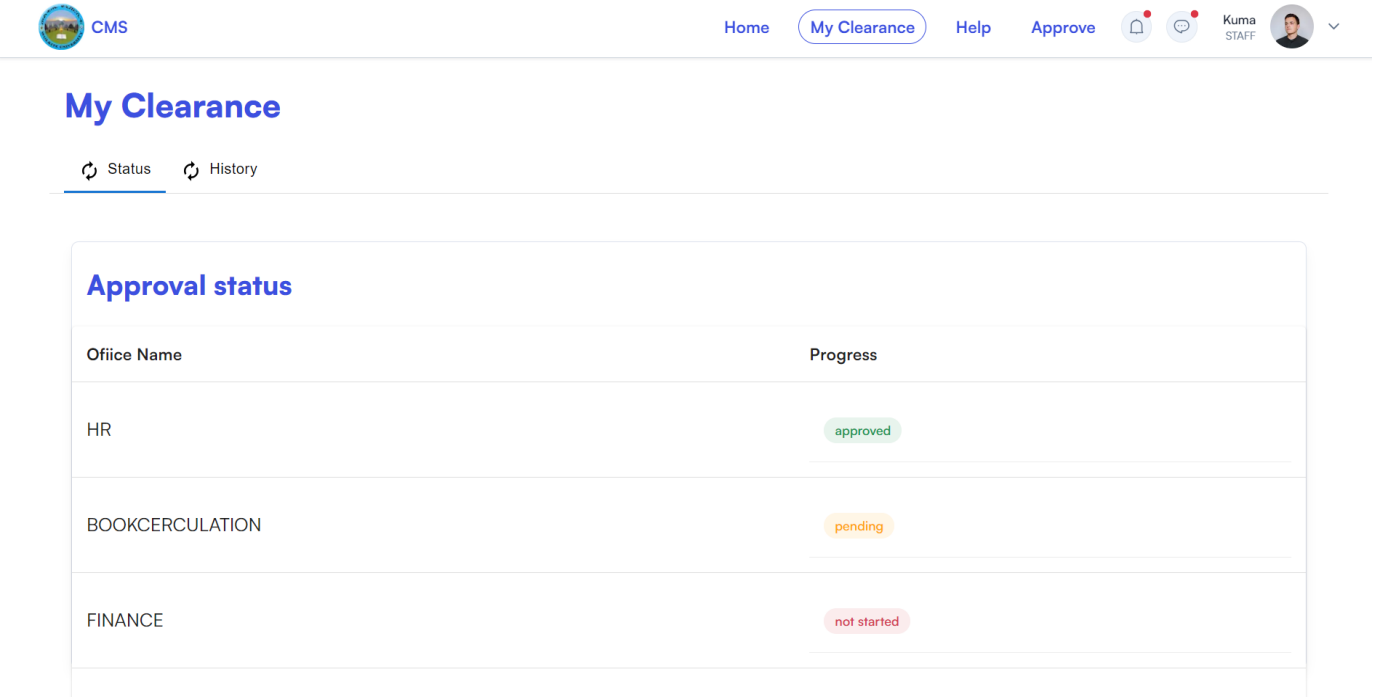


Figure 27. Track clearance status page of wkucms

Chapter 6

6. Implementation And Testing

6.1. Implementation of the Database

We opted for MongoDB, a NoSQL document-oriented database, as our primary data storage solution. This choice aligns well with the project's requirements for:

- **Flexible Schema:** Our data model is expected to evolve over time, and MongoDB's schema-less nature allows for easy adaptation without rigid table structures.
- **High Performance:** MongoDB excels in handling large datasets and frequent reads/writes, which is crucial for our application's anticipated user base and data volume.
- **Horizontal Scalability:** As our application grows, MongoDB can be readily scaled horizontally by adding more servers to distribute the data load.

6.1.1. Database Design

- **Creating Collections (Tables)**:
  - We'll create collections in MongoDB to represent the persistent models identified in the design document. Each collection will have fields corresponding to the attributes of the respective model.
  - Primary keys will be automatically generated by MongoDB's **ObjectId**.
  - Foreign key relationships can be implemented using references between documents in different collections.
  - We'll ensure that each collection satisfies at least the third normal form by organizing data into separate collections based on their entity types and avoiding redundant data.

❖ **Normalization:**

- While the concept of normal forms (like 3rd Normal Form) doesn't directly translate to NoSQL databases, we ensure **data integrity** by carefully designing our documents to minimize redundancy and maintain consistency.

❖ **Indexes:**

- Indexes are created on frequently queried fields within collections to significantly improve query performance. These act like signposts within the data, enabling MongoDB to locate relevant documents efficiently.

❖ **Views:**

- We will investigate the implementation of MongoDB views for frequently used complex queries. Standard views can simplify complex queries and potentially improve performance for read-heavy operations.

❖ **Triggers and Stored Procedures:**

- While MongoDB doesn't natively support triggers in the traditional RDBMS sense, we can explore alternative approaches to automate specific database operations using server-side Javascript functions. Stored procedures, however, are not directly applicable to MongoDB's architecture.

- MongoDB does not natively support **triggers** in the same way as RDBMS. Instead, developers can leverage MongoDB's capabilities to automate specific database operations using **server-side JavaScript functions.** These functions can be executed in response to certain events or conditions, providing a level of automation similar to triggers in RDBMS.

- **Stored procedures** are **not directly applicable** to MongoDB's architecture. MongoDB is a document-oriented NoSQL database that stores data in flexible, schema-less documents rather than tables with fixed schemas. As a result, the concept of stored procedures, which are tightly coupled with SQL and the relational model, does not directly translate to MongoDB.Instead of stored

procedures, developers working with MongoDB typically **encapsulate** complex data manipulation logic within application code. This approach allows for greater flexibility, as data manipulation can be tailored to specific application requirements and executed using MongoDB's query language and APIs.

❖ **Database Backup**:

- We'll configure a schedule for regular database backups to ensure data integrity and disaster recovery.
- We will establish a scheduled routine for regular database backups to uphold data integrity and facilitate disaster recovery procedures. For our projects, we employ MongoDB's **mongoDump** and **mongorestore** tools for executing backups and restorations.

❖ **Database Level Security**:

- We'll implement database-level security by configuring access control settings.
- Authentication mechanisms such as username/password authentication or MongoDB's native authentication mechanisms can be used.
- Role-based access control (RBAC) can be enforced to control access at the database level.

6.1.2. Tables as Persistent Models

- In MongoDB, representing **foreign keys** is slightly different from relational databases like SQL. MongoDB doesn't have built-in foreign key constraints like SQL databases. Instead, you typically represent relationships between documents by embedding documents or referencing documents from other collections.

```
const StepsSchema = new Schema({
 stepId: {
   type: String,
```

```
      required: true,
    },
    name: {
      type: String,

    },
    nextSteps: {
      type: [String],
      default: []
    },
    stepType: {
      type: String,
      required: true,

    },
  });
```

```
const OfficeSchema = new Schema({
 officeId: {
   type: String,
   unique: [true, "ID already exists!"],
   required: [true, "Id is required!"],
 },
 officeName: {
   type: String,
   required: [true, "officeName is required"],
 },

 password: {
```

```
    type: String,
  },

  location: {
    type: String,
    required: [true, "location is required"],
  },
  items: {
    type: String,
    required: [true, "items is required"],
  },
  type: {
    type: String,
    required: [true, "step type is required"],
  },
  status:{
    type:String,
    required:[true,"status is required"],
  }
});
```

```
const StudentRequestSchema = new Schema({
  userId: {
    type: String,
    required: [true, "Id is required!"],
  },
  reason: {
    type: String,
    required: [true, "Reason is required!"],
  },
  attachedFile: { type: String },
```

```
status: {
  type: [String],
  required: [true, "Status is required!"],
},
approvals: [
  {
    role: String,
    time: Date,
  },
],
rejections: {
  type: Array,
},
firstname: {
  type: String,
  required: [true, "firstname is required!"],
},
middlename: {
  type: String,
  required: [true, "middlename is required!"],
},
role: {
  type: String,
  required: [true, "role is required!"],
},
collegeName: {
  type: String,
  // required: [true, "collegeId is required!"],
},
departmentName: {
  type: String,
  // required: [true, "departmentId is required!"],
},
```

```
 _userId: {
   type: String,
 },
 blockNo: {
   type: String,
 },
 dateRequested: {
   type: String,
   required: [true, "Requested date is required!"],
 },
});
```

```
const UserSchema = new Schema({
 userId: {
   type: String,
   unique: [true, "ID already exists!"],
   required: [true, "Id is required!"],
 },
 firstname: {
   type: String,
   required: [true, "firstname is required"],
 },
 middlename: {
   type: String,
   required: [true, "middlename is required"],
 },
 lastname: {
   type: String,
   required: [true, "lastname is required"],
 },

 password: {
   type: String,
 },

 collegeName: {
   type: String,
 },
 departmentName: {
   type: String,
```

```
},
officename: {
  type: String,
},
image: {
  type: String,
},
year: {
  type: String,
},

role: {
  type: String,
},
privilege: {
  type: String,
},
email: {
  type: String,
},
profilePic: {
  type: String,
},
blockNo: {
  type: String,
},
status: {
  type: String,
  default: "active",
},
emailResetPassword: {
  type: String,
},
verificationCode: {
  type: String,
},
director: {
  type: String,
},
staffType: {
  type: String,
},

passwordResetTokenExpires: { type: Date, default: null },
```

```
 createdAt: {
   type: Date,
   default: Date.now,
 },
 updatedAt: {
   type: Date,
   default: Date.now,
 },
});
```

6.1.3. Defining indexes

```
const StepsSchema = new Schema({
   stepId: {
     type: String,
     required: true,
   },
   name: {
     type: String,
   },
   nextSteps: {
     type: [String],
     default: []
   },
   stepType: {
     type: String,
     required: true,
     index: true, // Adding an index on the stepType
field
   },
 });
```

```
const StudentRequestSchema = new Schema({
   userId: {
     type: String,
     required: [true, "Id is required!"],
     index: true, // Adding an index on the userId
field
   },
   reason: {
     type: String,
     required: [true, "Reason is required!"],
   },
   attachedFile: { type: String },
   status: {
     type: [String],
     required: [true, "Status is required!"],
   },
   approvals: [
     {
       role: String,
       time: Date,
     },
   ],
   rejections: {
     type: Array,
   },
   firstname: {
     type: String,
     required: [true, "firstname is required!"],
   },
   middlename: {
     type: String,
```

```
    required: [true, "middlename is required!"],
  },
  role: {
    type: String,
    required: [true, "role is required!"],
  },
  collegeName: {
    type: String,
    // required: [true, "collegeId is required!"],
  },
  departmentName: {
    type: String,
    // required: [true, "departmentId is required!"],
  },
  _userId: {
    type: String,
  },
  blockNo: {
    type: String,
  },
  dateRequested: {
    type: String,
    required: [true, "Requested date is required!"],
  },
});
```

6.1.4. Configuring Database-Level Security

Configuring database-level security involves implementing measures to control access to your database and ensure that only authorized users or applications can interact with it. Here are some steps to configure database-level security:

- **Authentication:**

Enable authentication mechanisms within our database system (e.g., MongoDB) to ensure that only authorized users can access the system.

Create user accounts for administrators, staff members, and other relevant roles, each with unique credentials.

Enforce strong password policies to prevent unauthorized access and ensure that passwords expire periodically.

- **Authorization:**

Define roles within the system, such as administrators, staff members, and students, each with specific permissions.

Grant appropriate privileges to each role based on their responsibilities and required operations within the clearance management system.

Avoid granting excessive permissions to minimize the risk of unauthorized access or misuse of system resources.

- **Encryption:**

  - User Passwords: Implement encryption mechanisms to secure user passwords stored in the database. Use strong cryptographic hashing algorithms (e.g., bcrypt) to encrypt passwords before storing them, ensuring that passwords are not stored in plaintext and are resistant to decryption attacks.

  - Staff Privileges: Encrypt sensitive data related to staff privileges, such as access levels or role permissions, to protect this information from unauthorized access or tampering. Utilize encryption techniques to safeguard staff privileges both at rest and in transit, ensuring that only authorized individuals can view or modify these settings

6.2. Implementation of the Class Diagram

```
const Role = {
   ADMIN: 'ADMIN',
   STUDENT: 'STUDENT',
   STAFF: 'STAFF',
 };
  export default Role;
```

```
class Status {
   constructor(studentStepData, adminStepData, academicStepData,
handleRequest) {
     this.studentStepData = studentStepData;
     this.adminStepData = adminStepData;
     this.academicStepData = academicStepData;
     this.handleRequest = handleRequest;
     this.userData = null;
     this.error = null;
     this.requestStatus = [];
     this.steps = null;
     this.router = useRouter();
   }
    fetchData = async () => {
     const response = await fetch("/api/userStatus");
     const data = await response.json();
     const updatedData = data.map((user) => ({
       ...user,
       id: user._id,
       roleId: user._id,
     }));
     this.userData = updatedData;
     this.handleError();
     this.processData();
   };
    handleError = () => {
```

```
    if (!this.userData && !this.error) {
      console.log("Loading...");
    }
    if (this.error) {
      console.error("Error fetching data:", this.error);
      console.log("Failed to fetch data");
    }
  };
   processData = () => {
    if (this.userData) {
      if (
        this.userData &&
        this.userData[0]?.staffType == "ACADEMIC"
      ) {
        let academicStep = {};
        this.academicStepData.forEach((data, index) => {
          academicStep[data.name] = data.nextSteps;
        });
        this.steps = academicStep;
      } else if (
        this.userData &&
        this.userData[0]?.staffType == "ADMIN"
      ) {
        let adminStep = {};
        this.adminStepData.forEach((data, index) => {
          adminStep[data.name] = data.nextSteps;
        });
        this.steps = { ...adminStep };
        delete this.steps.Director;
      } else {
        let studentStep = {};
        this.studentStepData.forEach((data, index) => {
          studentStep[data.name] = data.nextSteps;
        });
        this.steps = studentStep;
      }
       const currentStatus = this.userData[0]?.status;
      Object.keys(this.steps).forEach((key) => {
        const stepKey = key;
```

```
        const approvals = this.userData[0]?.approvals.map(
          (approval) => approval.role
        );
        const rejections = this.userData[0]?.rejections;
        let status = "Not Started";
         if (approvals && approvals.includes(key)) {
          status = "Approved";
        } else if (rejections && rejections.includes(key)) {
          status = "Rejected";
        } else if (
          currentStatus &&
          currentStatus.includes(key)
        ) {
          for (const element of currentStatus) {
            if (
              this.steps[element] &&
              (this.steps[element].includes(stepKey) ||
                (this.steps[this.steps[element][0]] &&
                  this.steps[this.steps[element][0]].includes(
                    stepKey
                  )))
            ) {
              status = "Not Started";
              break;
            }
            status = "Pending";
          }
        }
        this.requestStatus.push({ name: key, status: status });
      });
      this.setRequestStatus(this.requestStatus);
    }
  };
   handleReinitate = async (key) => {
    const response = await fetch(`/api/reinitiateRejected`, {
      method: "PATCH",
      body: JSON.stringify({
        reinitiate: this.requestStatus[key].name,
        objectId: this.userData[0]._id,
```

```
      }),
    });
    if (response.ok) {
      toast.success("Your clearance is reinitiated successfully");
    }
  };
   handlePrintClearance = () => {
    this.router.push(
      `/user/PrintClearance?clearanceId=${this.userData[0]?._id}`
    );
  };
}
```

6.3. Configuration of the Application Server

6.3.1. Application Server Selection Justification

We have chosen to utilize the **Next.js** framework for both the front and back end development of the clearance management system for Wolkite University. Next.js offers several advantages that align well with the requirements of the project:

- **Server-side rendering (SSR):** Next.js provides built-in support for SSR, which enhances the performance and SEO-friendliness of the application by rendering pages on the server before sending them to the client.
- **Efficient routing:** Next.js offers a straightforward and efficient routing system, simplifying navigation within the application.
- **Automatic code splitting:** With Next.js, code splitting is automated, allowing for faster page loads and improved performance.
- **Static file serving:** Next.js allows for the serving of static files, making it convenient for handling assets such as images, stylesheets, and client-side JavaScript files.
- **Ease of deployment:** Next.js applications can be easily deployed to various hosting platforms, including Vercel, AWS, and Heroku, among others.

6.3.2. Activities Performed

Proper Start and Shutdown of the Application Server:

- We have configured the Next.js application server to start and shutdown properly using commands provided by the framework's CLI (Command Line Interface). This ensures smooth initialization and termination of the server without any issues.

Organizing Folders and Files:

- We have organized the project's folders and files according to best practices recommended by Next.js, ensuring a structured and maintainable codebase. Components, pages, utilities, and other relevant files are logically grouped within the project directory.

Configuration for Local Access:

- The application server has been configured to work on the same machine, allowing easy access to contents locally for development and testing purposes. This configuration ensures seamless interaction with the application during the development phase.

Configuration for Remote Access:

- Additionally, we have configured the application server to allow access from remote machines, enabling stakeholders and users to interact with the system from different locations. Proper security measures have been implemented to safeguard the server from unauthorized access.

Separation of Application Server and Database:

- To adhere to best practices and ensure scalability and maintainability, we have separated the application server from the database layer. MongoDB is utilized as the storage solution for the application, while Next.js serves as the application server, facilitating a clean separation of concerns.

Configuration for Different Port Numbers:

- The server has been configured to work on different port numbers as required. This allows for flexibility in deployment and avoids conflicts with other services running on the same machine.

Conclusion:

In conclusion, the configuration of the application server for the clearance management system developed for Wolkite University adheres to best practices and ensures optimal performance, scalability, and security. By leveraging the Next.js framework for both front and back end development and utilizing MongoDB as the storage solution, we have created a robust and efficient system capable of meeting the university's requirements effectively.

6.4. Configuration of Application Security

6.4.1. Implementation of Input Validations

Input validation is a critical aspect of ensuring the security and integrity of the clearance management system. We have utilized the Yup validation library to implement comprehensive input validation mechanisms across all user inputs. Yup allows us to define schema-based validations for data entered by users, ensuring that only valid and expected data is accepted by the system. By enforcing strict input validation rules, we mitigate the risk of injection attacks such as SQL injection and cross-site scripting (XSS), enhancing the overall security posture of the system.

6.4.2. Encryption and Decryption Mechanisms

To safeguard sensitive data within the clearance management system, we have implemented encryption and decryption mechanisms. Encryption is applied to user passwords using strong hashing algorithms such as bcrypt before storing them in the database. Additionally, encryption is applied to staff privileges to protect sensitive access control information. By encrypting sensitive data, we ensure that even if unauthorized access occurs, the data remains unreadable and unusable, thereby enhancing the confidentiality and security of the system.

6.4.3. Clearly Defined Roles

Roles within the clearance management system are clearly defined to enforce access control and privilege management. The system distinguishes between three main roles: student, staff, and admin. Each role is assigned specific access privileges and responsibilities based on the user's role within the university hierarchy. By clearly defining roles, we ensure that users are granted appropriate permissions and access levels, minimizing the risk of unauthorized access and data breaches.

6.4.4. Assignment of Access Privileges

User accounts in the clearance management system are assigned with necessary access privileges based on their roles. Students have access to functionalities related to their academic clearance status and course registration. Staff members are granted additional privileges related to administrative tasks such as managing student records and clearance processes. Administrators have full access to system functionalities and administrative controls, enabling them to oversee and manage the entire clearance management system. By assigning access privileges according to roles, we enforce the principle of least privilege, limiting access to sensitive functionalities only to authorized users.

6.4.5. Implementation of Sessions

Sessions have been implemented within the clearance management system to maintain user authentication and session integrity. Upon successful login, a unique session token is generated for each user, which is used to authenticate subsequent requests. Session tokens are securely managed and validated with each request to ensure that only authenticated users can access system functionalities. Additionally, session timeouts are enforced to automatically log users out after a period of inactivity, reducing the risk of session hijacking and unauthorized access.

6.4.6. Compliance with Non-Functional Requirements

In addition to security measures, the clearance management system has been designed and developed to meet all non-functional requirements specified in the system features. This includes requirements related to performance, reliability, scalability, and usability. By adhering to non-functional requirements, we ensure that the system operates effectively, reliably, and securely, meeting the needs of Wolkite University and its stakeholders.

In summary, the clearance management system incorporates robust security measures, including input validation, encryption, role-based access control, session management, and compliance with non-functional requirements. By addressing these security aspects, we have created a secure and reliable system that protects sensitive data, mitigates security risks, and meets the high standards of security expected by Wolkite University.

6.5. Implementation of User Interface

The implementation of the user interface for the clearance management system adheres to user-centered design principles, with a focus on placing users in control, reducing memory load, and ensuring consistency. By prioritizing user needs, iteratively refining the design, and maintaining consistency across various aspects of the interface, we have created a user-friendly and intuitive system that meets the expectations of students, staff, and administrators at Wolkite University.

6.5.1. User-Centered Design

In the development of the user interface for the clearance management system using Next.js and MongoDB, we have prioritized user-centered design principles. Our aim is to place users in control of the interface, ensuring that their needs, preferences, and workflows are at the forefront of the design process. To achieve this, we have employed the following strategies:

- **User Research:** We conducted thorough user research to understand the requirements and expectations of the stakeholders, including students, staff, and administrators at Wolkite University. This helped us identify user personas, their goals, pain points, and preferences, which guided the design decisions.
- **Iterative Design Process:** We adopted an iterative design process, involving continuous feedback loops with users and stakeholders. By soliciting feedback early and often, we iteratively refined the user interface to better align with user needs and preferences.
- **User Testing:** We conducted user testing sessions to evaluate the usability and effectiveness of the interface. This allowed us to identify usability issues, accessibility concerns, and areas for improvement, which were addressed iteratively to enhance the overall user experience.

6.5.2. Reduction of User Memory Load

To reduce the cognitive burden on users and minimize their memory load, we have implemented several design strategies within the user interface:

- **Clear Navigation:** We have designed a clear and intuitive navigation system that enables users to easily navigate between different sections of the application. Consistent navigation patterns and logical grouping of menu items help users quickly locate the desired functionality without relying on memory.
- **Contextual Feedback:** We provide contextual feedback and guidance to users throughout their interaction with the system. This includes informative error messages, tooltips, and inline help text that assist users in understanding their actions and correcting errors without having to rely solely on memory.
- **Streamlined Workflows:** We have optimized user workflows to minimize the number of steps required to accomplish common tasks within the clearance management system. By eliminating unnecessary complexity and reducing cognitive overhead, we ensure that users can complete tasks efficiently and with minimal cognitive strain.

6.5.3. Consistency in User Interface

Consistency is a key principle in ensuring a cohesive and intuitive user experience. In the implementation of the user interface for the clearance management system, we have focused on maintaining consistency across various aspects of the interface:

- **Visual Consistency:** We have established consistent visual elements such as color schemes, typography, iconography, and layout grids throughout the application. This creates a unified visual language that helps users navigate the interface with ease and familiarity.
- **Interaction Consistency:** We have standardized interaction patterns and behaviors across different components and screens of the application. Consistent interaction cues, such as button styles, form elements, and navigation patterns, ensure that users can predict the outcome of their actions and navigate the interface confidently.

- **Content Consistency:** We have maintained consistency in terminology, language, and content presentation across the user interface. This ensures clarity and coherence in communication, reducing ambiguity and cognitive load for users.

6.6. Testing

6.6.1. Test Case

❖ **User Registration:**

- Test Case: Verify that users can register successfully with valid credentials.
- Test Steps: Enter valid user information (e.g., username, email, password) and submit the registration form.
- Expected Result: User registration is successful, and a confirmation message is displayed.

❖ **User Update:**

- Test Case: Ensure that users can update their profile information.
- Test Steps: Access the user profile page, modify the user information, and save the changes.
- Expected Result: User profile is updated successfully with the new information.

❖ **User Deactivation:**

- Test Case: Verify that administrators can deactivate user accounts.
- Test Steps: Access the admin dashboard, locate the user account to deactivate, and perform the deactivation action.
- Expected Result: User account is deactivated, and the user loses access to the system.

❖ **Clearance Approval:**

- Test Case: Ensure that the staff can approve clearance requests.

- Test Steps: Access the clearance approval interface, review pending clearance requests, and approve a request.
- Expected Result: Clearance request is approved, and the corresponding user is notified of the approval status.

❖ **Clearance Rejection:**

- Test Case: Verify that the staff can reject clearance requests.
- Test Steps: Access the clearance approval interface, review pending clearance requests, and reject a request.
- Expected Result: Clearance request is rejected, and the corresponding user is notified of the rejection status.

❖ **Messaging and Notification:**

- Test Case: Ensure that users receive messages and notifications.
- Test Steps: Send a message/notification from one user account to another or from the system to a user.
- Expected Result: The recipient receives the message/notification promptly and can view it in their inbox or notification panel.

❖ **View Clearance Status:**

- Test Case: Verify that users can view their clearance status.
- Test Steps: Access the clearance status page and view the current status of clearance requests.
- Expected Result: Users can see the status of their clearance requests (e.g., pending, approved, rejected) accurately.

❖ **Clearance Request Submission:**

- Test Case: Ensure that users can submit clearance requests.

- Test Steps: Access the clearance request form, fill in the required information, and submit the request.
- Expected Result: The clearance request is submitted successfully, and users receive a confirmation message.

6.6.2. Testing Tools and Environment

- **Testing Tools:** We utilize testing frameworks such as Jest and React Testing Library for unit and integration testing, as well as tools like Cypress for end-to-end testing.
- **Environment:** Testing is conducted in a controlled environment that mimics the production setup, including servers hosting the Next.js application and MongoDB database. We use development and staging environments for testing purposes to ensure the reliability and accuracy of test results.

6.6.3. Unit Testing

Unit testing focuses on testing individual units or components of the system in isolation. We use Jest and React Testing Library to write and execute unit tests for React components, API endpoints, and utility functions. Mocking libraries are utilized to simulate external dependencies and ensure tests are deterministic and repeatable.

6.6.4. System Testing

System testing involves testing the system as a whole to validate its compliance with functional and non-functional requirements. We perform system testing by executing end-to-end tests using tools like Cypress(end-to-end testing framework primarily used for testing web applications. It's designed to simplify the process of writing, running, and debugging tests by providing a comprehensive set of features and a user-friendly interface. Cypress is particularly popular for its ability to conduct automated tests that simulate user interactions within the browser environment). Test scenarios cover user journeys, including user registration, clearance request submission, approval workflows, and messaging/notification functionalities.

6.6.5. Integration Testing

Integration testing focuses on testing the interactions and integration points between different components/modules of the system. We conduct integration testing to ensure seamless

communication and data flow between frontend and backend components, as well as third-party integrations such as sending emails or SMS notifications. Mocking and stubbing techniques are employed to isolate components and simulate interactions.

6.6.6. Acceptance Testing

Acceptance testing involves validating the system's compliance with user requirements and expectations. We collaborate closely with stakeholders, including students, staff, and administrators, to define acceptance criteria and scenarios. User acceptance testing (UAT) sessions are conducted to allow stakeholders to interact with the system and provide feedback on its usability, functionality, and overall satisfaction. Any identified issues or enhancements are addressed iteratively to ensure the system meets stakeholder needs effectively.

Chapter 7

7. Conclusion And Recommendation

7.1. Conclusion

The implementation of an Automated Clearance Management System at Wolkite University marks a significant step towards streamlining clearance processes and enhancing overall efficiency. By transitioning from a manual to an automated system, the university aims to address the inherent challenges associated with manual processing, including manpower usage, errors, and time consumption. Through a structured approach, the project has identified key stakeholders, elicited their requirements, and documented them systematically. The synthesized requirements serve as the foundation for the development of the Automated Clearance Management System, which promises to revolutionize the clearance approval experience for requesters. Importantly, the conclusion reaffirms the purpose of the project, emphasizing its role in resolving existing issues within the manual clearance system and highlighting its differentiation from related projects. Looking ahead, opportunities for future project works are identified, signaling a commitment to continuous improvement and innovation in addressing evolving needs and challenges.

In addition to streamlining clearance processes and enhancing efficiency, the implementation of the Automated Clearance Management System underscores the transformative impact of technology on administrative operations within higher education institutions. By leveraging automation and digitalization, Wolkite University aims to modernize its clearance procedures, aligning with broader trends towards digitization in educational administration. The project's systematic approach to requirements gathering, analysis, and validation demonstrates a commitment to delivering a solution that meets the specific needs and challenges faced by the university. Furthermore, the adoption of best practices in software development, including agile methodologies and user-centered design principles, ensures that the Automated Clearance Management System is not only functional but also user-friendly and intuitive. As a result, the implementation of this system represents a paradigm shift in how clearance processes are conducted, marking a significant milestone in the university's journey towards digital transformation and operational excellence.

Although our project aims to modernize and streamline clearance processes through automation, it's imperative to recognize its inherent limitations. Notably, the system doesn't cover the registration of records and associated properties, which means that record verification will still require manual intervention outside the system's scope. This limitation underscores the importance of users and stakeholders being cognizant of the system's boundaries when utilizing the Automated Clearance Management System.

7.2. Recommendation

In light of the project's findings and the implementation of the Automated Clearance Management System at Wolkite University, the following recommendations are proposed:

- **Expansion of System Scope:** Considering the identified limitation regarding record registration, it is recommended to explore opportunities for expanding the system's scope to include the registration of records and associated properties. By integrating record registration functionalities, the system can offer a more comprehensive solution for managing clearance processes, reducing the need for manual interventions.

- **Enhanced Integration:** To further optimize efficiency and streamline operations, it is recommended to explore possibilities for enhanced integration with existing university systems and databases. Seamless integration with student databases, employee records, and other relevant systems can facilitate the retrieval and verification of information, improving the overall clearance validation process.

- **Continuous Improvement:** As with any technology solution, it is essential to prioritize continuous improvement and refinement of the Automated Clearance Management System. This includes soliciting feedback from users and stakeholders, monitoring system performance, and implementing iterative updates and enhancements based on evolving needs and technological advancements.

- **Training and Awareness Programs:** To ensure the successful adoption and utilization of the system, it is recommended to develop comprehensive training and awareness programs for users and stakeholders. Training sessions, user guides, and support resources can help familiarize individuals with the system's functionalities, promote best practices, and address any concerns or challenges encountered during implementation.

- **Regular Evaluation and Assessment:** Establishing mechanisms for regular evaluation and assessment of the Automated Clearance Management System's effectiveness and impact is essential. Conducting periodic reviews, surveys, and performance evaluations can provide valuable insights into the system's strengths, weaknesses, and areas for improvement, enabling informed decision-making and optimization efforts.

These recommendations are proposed to guide future actions and initiatives aimed at maximizing the benefits and effectiveness of the Automated Clearance Management System at Wolkite University. By embracing these recommendations, the university can continue to advance its clearance processes, enhance operational efficiency, and deliver an exceptional user experience for students, staff, and stakeholders.

References

- Software Engineering, Ian Sommerville, October 2009, 9th edition
- WKU Industrial Project Guideline, College of Computing and Informatics, May 2019, version 2
- WKU student clearance workflow forms, WKU main registrar office
- WKU academic staff clearance workflow forms, WKU human resource management directorate director, January 2024
- WKU admin staff clearance workflow forms, WKU human resource management directorate director, January 2024

Appendix

Interview questions

General Questions

- What is the full name of the office (in English and Amharic)?
- Who are the key stakeholders and users?
    - ✓ Do their goals differ?
    - ✓ If so, how?
- What are the problems of the current manual system?
- Did you think Automating the current system will solve your problems? If yes how?
- Is there any existing system documentation or any other file which is related with your work? If so, where? And Who else should I talk to?
- What seems like the over all process during serving the students in your office?

Current Needs

- What information or data do you need from this system that you don't have now?
- How do you store and organize data and information in the current system? Is it need improvement or not?
- Is any of this data currently captured in any other system?
- Is the data and/or functionality that you use currently shared by other systems or users?

Current problems

- What problems do you face during your work that is related to clearance?
- Do you have performance problems that need to change? tell us about that?
- Are there work processes that are missed due to the boring working process of the manual system? What are they?
- Is there frequently faced process during your work?

Criteria for success

- Are there any prerequisite processes that must be succeed before coming to your office?
- Where the student has accomplished the process in your office?
- Are there other data that do you use from other system? What are that data?
- Does the officer who worked in this process has enough computer skills to use automated system?

Improving an Existing System

- Will the new system have additional functionality?
- Will the new system help you be more efficient?
  - ✓ To what extent?
- What is most important (rank in order of importance):
  - ✓ Application is easier to use
  - ✓ Application has nicer front-end
  - ✓ Application has additional functionality (list)
  - ✓ Application is more efficient
  - ✓ Application is redesigned to better reflect the business (The application has been updated to properly represent the company).

High-Level Functions

- What will this project/system do that is entirely new (What entirely new thing will this project/system do?)?
- Does the current system do things that this system will not do?
- How do these functions interact with each other?
- Are there other systems this system will interface with?