People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
M'hamed Bougara University Boumerdes



Faculty of Sciences
Department of Computer Science

Field : Mathematics and Computer Science

Program : Computer Science

Specialization: Computer Systems

N° de l'Arrêté d'habilitation de la spécialité: arrêté n°872 du 26/07/2016

End-of-Studies Dissertation for the Award of the Bachelor's Degree (Licence Académique)

Topic

Design and Implementation of a Web Application for Workflow Automation and Secure Document Sharing in Enterprise Environments

Presented by:
DEGHMOUM NOUHA AMANI
KAHLOUCHE MERIEM
NEKKAA NOURHANE

Internship carried out at: GCB Company in BOUMERDES.

Defended on june2026 Before the jury composed of:
IMINE BELKCEM : Examiner
BELKASMI DJAMEL : Supervisor
AOUCHICHE SOFIANE : External Supervisor



Thanks

This work is a concrete testimony of our perseverance. We dedicate it with great pride to all our loved ones.

To our dear parents

With our deepest gratitude for your unconditional love, your sacrifices, and your unwavering support throughout this journey. No words can truly express the respect and love we carry for you.

To our sisters and brothers

For your constant presence, your encouragement, and every shared moment that kept us motivated.

Nouha, Nourhane and Meriem

We are pleased to express our sincere gratitude to all those who contributed to the success of this project.

First and foremost, Alhamdulillah for granting us the patience and strength to accomplish this work.

We extend our deepest thanks to our parents and siblings for their continuous support and encouragement throughout this journey.

Our warmest gratitude goes to Mr. BELKASMI Djamal, our academic supervisor, and to Mr. AOUCHICHE Sofiane, our internship supervisor, as well as the entire team at GCB spa, for their guidance, availability, and professionalism.

Finally, we thank every person who, near or far, contributed in any way to making this project a reality.

Abstract

This project focuses on the design and implementation of a secure web application for GCB, a subsidiary of SONATRACH, aimed at digitalizing its administrative workflows and document sharing. Built on a three-tier architecture (React, Laravel, MySQL), the platform replaces a slow, paper-based system with an automated, traceable alternative. Security and access control are handled via LDAP integration with Active Directory and a Role-Based Access Control (RBAC) system for Employees, Secretaries, and Administrators. The application enables digital submissions for leaves, mission orders, and vehicle requests with real-time notifications, alongside a 1GB secure personal storage space for each user. Modelled with UML and Merise, the final solution significantly optimizes administrative processing time and organizational productivity.

Key words: Web application, workflow automation, document management, LDAP, Laravel, React, MySQL.

Résumé

Ce projet concerne la conception et la réalisation d'une application web pour la GCB, filiale de SONATRACH, visant à digitaliser les flux administratifs et le partage de documents. Basée sur une architecture trois tiers (React, Laravel, MySQL), la plateforme remplace un système manuel papier lent par une alternative automatisée et traçable. La sécurité est assurée par une intégration LDAP avec l'Active Directory et un contrôle d'accès basé sur les rôles (RBAC) pour trois profils : Employé, Secrétaire et Administrateur. Le système permet la soumission numérique des congés, ordres de mission et demandes de véhicules avec notifications en temps réel, tout en offrant un espace de stockage sécurisé de 1Go par utilisateur. Modélisée avec UML et Merise, cette solution optimise le temps de traitement administratif
et la productivité.

Mots-clés : Application web, automatisation, gestion documentaire, LDAP, Laravel, React, MySQL, Active Directory.

ملخص

يتناول هذا المشروع تصميم وتطوير تطبيق ويب مخصص لشركة GCB ، التابعة لمجمع سوناطراك، بهدف رقمنة المسارات الإدارية ومشاركة الوثائق. يعتمد الحل على معمارية ثلاثية الطبقات React و Laravelو MySQL كبديل للنظام الورقي التقليدي لضمان سرعة المعالجة وتتبع الملفات. تم تأمين النظام عبر دمج بروتوكول LDAP مع الـ Active Directory وتطبيق نظام صلاحيات (RBAC)لثلاثة أدوار: موظف، سكرتير، ومسؤول. يتيح التطبيق تقديم طلبات العطل، أوامر المهمة، وطلبات المركبات رقمياً مع إشعارات فورية، بالإضافة إلى توفير مساحة تخزين آمنة بسعة 1 جيجابايت لكل مستخدم. تم نمذجة المشروع باستخدام UML وMerise، مما أسفر عن منصة فعّالة قلّصت وقت المعالجة الإدارية ورفعت الإنتاجية العامة.

الكلمات المفتاحية: تطبيق ويب، أتمتة سير العمل، إدارة الوثائق LDAP، Laravel، MySQL، Active Directory، React .

Table list:

Table 2.1: Functional Requirements 12
Table 2.2: Non-Functional Requirements 13
Table 2.3: Textual description of use case “authenticate” 14
Table 2.4: Textual description of use case” submit request” 15
Table 2.5: Textual description of use case “share document” 15
Table 2.6: Textual description of use case” archive document” 16
Table 4.7: Specification machine 1 36
Table 4.8: Specification machine 2 36
Table 4.9: Specification machine 3 36
Table 4.10: Software environment 37
Table 4.11: Technology stack 38

Figure list:

Figure 1.1: GCB organizational chart 6
Figure 2.2: Global use case Diagram 13
Figure 2.3: Global sequence diagram 18
Figure 2.4: Authenticate sequence diagram 19
Figure 2.5: Submit Request sequence diagram 20
Figure 2.6: Share Documents sequence diagram 21
Figure 2.7: Archive Documents sequence diagram 22
Figure 3.8: Architecture diagram 25
Figure 3.9: Class diagram 27
Figure 3.10: Conceptual Data Model (MCD) 29
Figure 3.11: Logical Data Model (MLD) 30
Figure 3.12: Data Base Schema 31
Figure 4.13: Deployment diagram 34
Figure 4.14: Login page 36
Figure 4.15: Employee Dashboard 37
Figure 4.16: New Request 37
Figure 4.17: Example of request ''Leave Request'' 38
Figure 4.18: Example of request ''Vehicle Request'' 38
Figure 4.19: History Requests 39
Figure 4.20: Drive Space 39
Figure 4.21: Archive Documents 40
Figure 4.22: Shared with me 40
Figure 4.23: Shared with people 41
Figure 4.24: Employee Notifications 41
Figure 4.25: Secretary Dashboard 42
Figure 4.26: Incoming Requests 42
Figure 4.27: Secretary Notifications 43
Figure 4.28: Admin Dashboard 43
Figure 4.29: User management 44
Figure 4.30: Group management 44
Figure 4.31: System log 45
Figure 4.32: Statistics 45

Abreviation List:

GCB: National Civil Engineering and Building Company.
AD: Active directory.
IT: Information technology.
LDAP: Lightweight Directory Access Protocol.
MCD: Conceptual Data Model.
MLD: Logical Data Model.
OMG: Object Management Group.
RBAC: Role-Based Access Control.
SPE: Strategy, Planning and Research Directorate.
SQL: Structured Query Language.
UML: Unified Modeling Language.
HTTP: Hypertext Transfer Protocol.
API: Application Programming Interface.

Summary

General introduction .....

Chapter 01: .....

Presentation of the host organization and project context.....

introduction..... 3

1. Presentation of the host organization (GCB): 3

1.1 : Presentation of GCB..... 3

1.2 : organization of GCB 3

1.3 : GCB organizational chart..... 4

2. IT Department 4

2.1 : Presentation of the IT Department 4

3. Project context 5

3.1 : Project presentation 5

3.2 : Issues: 5

3.3 : Proposed solution..... 5

3.4 : Objectives: 6

Conclusion 6

Chapter 02:..... 7

Requirement Analysis..... 7

Introduction..... 8

1. Presentation UML (Unified Modelling Language): 8

2. Analysis and Specification of requirements: 8

2.1. Description of the Current System (Manual procedures)..... 8

2.2. Functional Requirements:..... 9

2.3. Non-functional Requirements: 8

2.4. Identification of actors:.....10

3. Use Case Diagrams: 10

3.1. Global Use Case diagram 10

3.2. Textual description of use case: 11

4. sequence diagrams:..... 15

4.1. Global sequence diagram 15

4.2. sequence diagram of the use case “authenticate “:..... 16

4.3. sequence diagram of the use case “submit request “:..... 16

4.4. sequence diagram of the use case “share documents” 17

4.5. sequence diagram of the use case “archive documents” 18

Conclusion 19

Chapter 03: 20
System Design 20
introduction 21
1. System architecture: 21
1.1. General architecture (3-Tier Architecture): 21
1.2. Architecture Diagram 22
2. Static Modelling (UML): 23
2.1. Class Diagram 23
3. Data Base Design 25
3.1. Conceptual Data Model (MCD): 26
3.2. Logical Data Model (MLD): 27
3.3. Data Base Schema 29
conclusion 29
Chapter 04: 30
System Realization and Implementation 30
introduction 31
1. Deployment Diagram 31
2. Development Environment 33
2.1. Hardware environment 33
2.2. Software environment 34
2.3. Technology Stack: 35
3. Implementation of core module: 36
3.1. Authentication and directory integration 36
3.2. Cloud Storage Integration 36
4. Presentation of the graphical interfaces 37
4.1. Authentication page 36
4.2. Employee Space 36
4.3. Secretary Space 43
4.4. Admin Space 44
conclusion 47
General conclusion 48

General introduction

In the era of digital transformation, modernizing internal administrative processes is crucial for organizations to ensure operational efficiency, traceability, and security. GCB, a prominent Algerian national company and subsidiary of SONATRACH, faces significant challenges due to its reliance on manual, paper-based workflows for routine administrative requests (such as leave and mission orders). This traditional approach leads to processing delays, risks of document loss, and a lack of real-time visibility. Furthermore, the absence of a centralized document management system creates operational bottlenecks and security risks across its regional units.

To address these inefficiencies, this graduation project (PFE) focuses on the design and implementation of a centralized, secure, and role-aware web application. Developed in collaboration with the IT Department of GCB's Strategy, Planning, and Research Directorate (SPE) in Boumerdes, the proposed solution replaces manual workflows with a digital platform built on a robust three-tier architecture: a React.js frontend, a Laravel 11 REST API backend, and a MySQL database. The system delegates user authentication to the company's existing Active Directory via the LDAP protocol and enforces Role-Based Access Control (RBAC) across three distinct roles (Employee, Secretary, and Administrator). Additionally, it integrates secure document management features, including a 1 GB personal storage quota and tokenized, time-limited sharing mechanisms.

This report is structured into four chapters documenting the project's complete lifecycle:

Chapter 1: presents the host organization, establishes the project context, and outlines the problem statement and objectives.

Chapter 2: analyzes the system requirements through UML use case and sequence diagrams.

Chapter 3: details the technical design, including the architectural layout, UML class diagrams, and Merise database modeling.

Chapter 4: describes the implementation environment, core technical modules, and showcases the final application interfaces.

Chapter 01:

Presentation of the host organization and project context.

introduction:

This chapter presents the host organization, GCB spa, with a focus on its IT department where the internship took place. It then presents the project context, identifies the key problems that motivated this work, and outlines the proposed solution and its objectives.

1. Presentation of the host organization (GCB):

1.1 : Presentation of GCB:

GCB (National Civil Engineering and Building Company) is a leading construction company in Algeria and a subsidiary of SONATRACH, operating within the oil and gas sector. It is mainly involved in civil engineering and construction projects, such as infrastructure development, pipeline construction and industrial projects. The company has about 6,000 employees and has a share capital of 40 billion Algerian Dinars, which places it among the leading construction companies.

The internship for this project was carried out within the Strategy, Planning and Research Directorate (SPE), specially in its IT Department, located at the company's headquarters in Boumerdes.

1.2 : organization of GCB:

The company is organized into regional units that carry out various activities, supported by specialized central technical departments such as the departments including the Engineering and Architecture Directorate. This organization helps it respond efficiently to regional needs while maintaining consistent standards across its operations.

Administrative activities such as leave requests, mission orders, and vehicle usage are still carried out manually, requiring employees to fill out paper forms and submit them directly to the department secretary.

1.3 : GCB organizational chart:

This diagram explains the internal structure of the company, from general management to the regional offices.

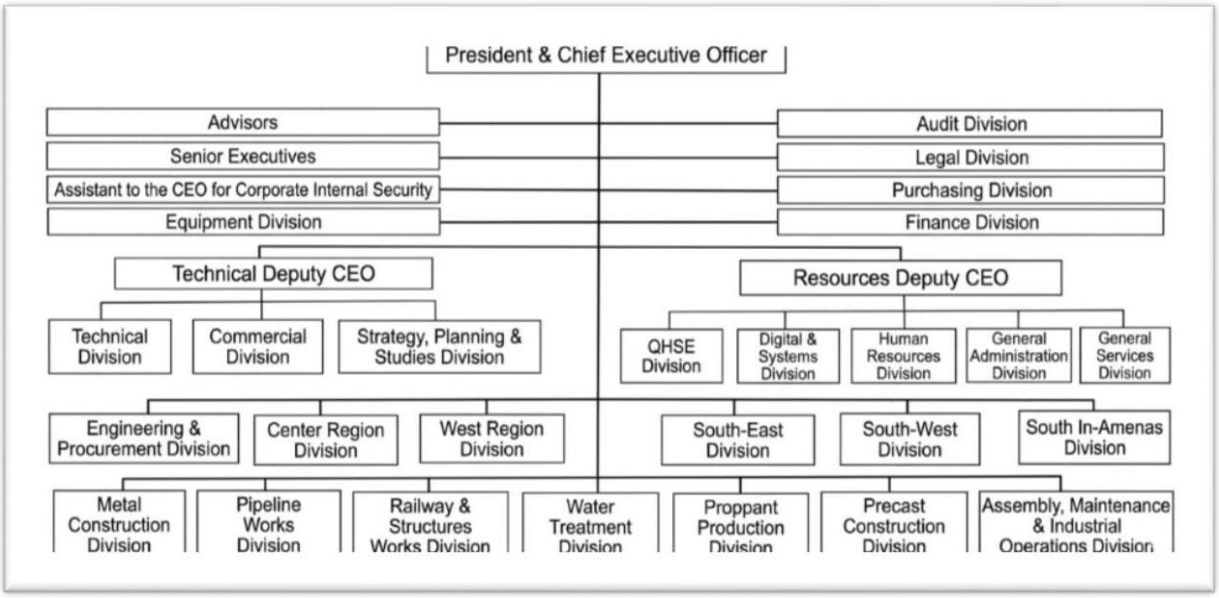


Figure 1.1: GCB organizational chart

2. IT Department:

2.1 : Presentation of the IT Department:

The IT Department part of the SPE Directorate, is in charge of managing and developing GCB 's information systems and infrastructure, while actively driving the company's digital transformation.

Currently, administrative processes depend on basic tools like e-mail and manual procedures. Requests are handled through non-centralized methods, which can lead to delays, lack of traceability, and difficulties in accessing information.

IT Department is responsible for managing the company's information system, network infrastructure, and computer equipment. It is structured around four roles:

- **IT Director:** oversees IT strategy, resource planning, and alignment with the company's technological needs.
- **Head of IT Department:** manages daily operations, resources, and project execution.
- **IT Engineers:** creating, maintaining and developing of the company's information systems and software solutions.

- **Senior Technicians:** ensure the daily operational maintenance of hardware, software, and network components.

3. Project context:

3.1 : Project presentation:

In the context of GCB's commitment to digital modernization, this project focuses on the digital overhaul of internal workflows.

This project consists of designing and developing a web application dedicated to GCB that allows employees to submit administrative requests and manage document sharing in a centralized and secure environment, Replacing the current paper-based process, and enabling GCB employees to interact with the administration digitally.

3.2 : Issues:

The current administrative system at GCB is based on manual procedures and non-centralized tools, which leads to several critical issues:

- **Processing delays:** Managing request requires manual handling and transfer between departments, which slows down response time.
- **Risk of data loss and errors:** Paper documents can be misplaced, filled out incorrectly.
- Lack of traceability: Information is stored in various environments, making it hard to access and track.
- **Security concerns:** Inefficient document management and uncontrolled file sharing may expose sensitive data.

These challenges negatively impact organizational efficiency and lead to higher operational costs and unreliable data management. This leads us to the following research question: "How can a secure and efficient system to manage administrative workflows and document sharing in a large-scale enterprise?"

3.3 : Proposed solution:

To handle these challenges at GCB, this project proposes the development of a three-tier web application architecture composed of:

- Presentation tier: User interface accessible via web browsers.
- Application tier: Handles business logic, request processing, and access control.
- Data tier: Stores and manages information that is processed by the application layer securely.

The system uses LDAP with Active Directory for central identity control, ensuring that only approved users can log in.

LDAP connects the system to the company's infrastructure, and it centralizes user management.

Security and Access Control:

- Role-based access control (RBAC) will be implemented (Administrator, Employee, Secretary).
- Secure authentication and session management.
- Limitation on file access and controlled sharing.
- Protecting sensitive files.

Document Management (cloud-based storage):

- Each user will have a 1GB storage quota by default, defined based on typical organizational needs and to optimize storage resources.
- Secure file sharing via controlled access or generated secure, time limited links.

3.4 : Objectives:

The main goal of this project is to design and build a secure, efficient, and adaptable web application that improves administrative workflows and document management within GCB.

The system is created to:

- Reduce administrative request processing time by at least 50% compared to manual procedures.
- Eliminate the use of paper-based requests (by 90%) of administrative workflows at GCB.
- Centralize 100% of documents within a unified and structured platform.
- Ensure full (100%) traceability of user actions and requests.
- Improve data security by applying authentication, role-based permissions, and secure file sharing methods.
- Manages storage by setting a controlled quota (1GB per user by default) and tracking usage.

Conclusion:

In this chapter, we have provided a global overview of GCB, and its IT department. It also introduced the context of the project and the problems that motivated its development. The next chapter will focus on the analysis of requirements and the detailed study of the proposed system.

Chapter 02:

Requirement Analysis

Introduction:

After presenting the project context, this chapter defines the system's requirements and uses UML to illustrate how users will interact with the application. It also describes the functional and non-functional needs of the project.

1. Presentation UML (Unified Modelling Language):



To model the system's requirements and structure, we use the Unified Modelling Language (UML), standardized by the Object Management Group (OMG). The diagrams used in this chapter include: Use Case diagrams to represent actor interactions, Sequence diagrams to model process flows, and a Class diagram to describe the static structure of the system.

2. Analysis and Specification of requirements:

2.1. Description of the Current System (Manual procedures):

At GCB, all the administrative requests are handled on paper. Employees fill out forms by hand and deliver them to the secretary, who then obtains signatures from the department head and director. This creates delays, missing documents, and no visibility into a request's status. Furthermore, document sharing between employees is completely disorganized, with no controlled or secure way to share files.

The proposed web application addresses these limitations. Employees can submit requests digitally, with automatic routing to the right people for approval. It also presents a secure shared document space, where employees can share files directly with selected colleagues (only those granted access can view them) and archive any documents they no longer need.

2.2. Functional Requirements:

Functionality	The System Shall
Authentication	- allow users to LOGIN using a username and password. - verify user credentials before access. - allow users to log out securely.
Request Management	- allow users to submit different types of requests (annual leave, recovery leave, mission order, vehicle request). - store requests in the database. - allow users to view and track their submitted requests. - allow the secretary to receive and access requests. - notify the secretary when a new request is submitted. - notify the secretary when a request is deleted(canceled). - allow users to delete their requests. - allow secretary to print requests.
Document Management	- allow users to upload documents. - allow users to share documents with other users either by generating a secure link or by selecting specific recipients. - allow users to view shared documents. - allow documents to be archived, unarchived, downloaded and accessed by user.
User Management	-allow the administrator to activate or disactivate a user account. -allow the administrator to create groups. -allow the administrator to set roles. -allow the administrator to set quota. -allow the administrator to set users to a group. -allow the administrator to consult logs

Table 2.1: Functional Requirements

2.3. Non-functional Requirements:

Non-functional Requirements	Specification
Performance	API response time under 800ms for standard requests, fast "Vite-powered" frontend for sub -200ms page transitions.
Security	LDAP authentication, HTTPS for all traffic, bcrypt password hashing, token-based session management via Laravel.
Scalability	The backend built on Laravel, must allow for future functional integrations without restructuring the existing codebase.
Reliability	The 1GB per-user quota must be strictly enforced server-side on every upload request, with 422 errors returned on violation.
Portability	The backend must operate within a virtualized environment (Hyper-V / Windows Server), and the frontend must run in any modern browser (example: chrome).

Table 2.2: Non-Functional Requirements 1

2.4. Identification of actors:

Employee: Submits requests (Annual leave, Recovery leave, Mission orders, and vehicle requests). Receives an automated acknowledgment of Receipt (accusé de reception) upon submission. Manages a personal Client Space" DRIVE" with 1 GB for documents storage and sharing. Consults their own request history and document status.

Secretary: Monitors the digital dashboard to process employee submissions. Manages for physical signature workflows. Executes the final document archiving. Serves as the operational link between the digital platform and GCB 's physical administration.

Administrator: Manages user accounts (creation, edits user information, delete) and role assignments and Active Directory (LDAP) synchronization to manage active accounts.
Manages global storage quotas, and ensure data security.

3. Use Case Diagrams:

3.1. Global Use Case diagram:

The UML use case diagram defines the functional boundaries of a document and request management system through three actors: Employee, Secretary, and Admin, and their associated use cases. A generalization relationship dictates that the Secretary actor inherits all use cases of the Employee, while the admin actor remains functionally separate, interacting exclusively with system administration use cases such as *Manage Drive Space*, *Manage Users*, and *View System Log*. The diagram maps explicit dependencies using standard UML stereotypes: the *Authenticate* use case serves as a universal prerequisite, connected via mandatory <<include>> relationships from almost all primary use cases across all actor lanes. Furthermore, conditional behavioral variations are captured through

<<extend>> and <<include>> chains linked to document sharing and downloading, specifically connecting *Share Document with Recipient* to *Verify Access Right*, *Generate Secure Link*, *Generate Download Link*, and *Link Validity*.

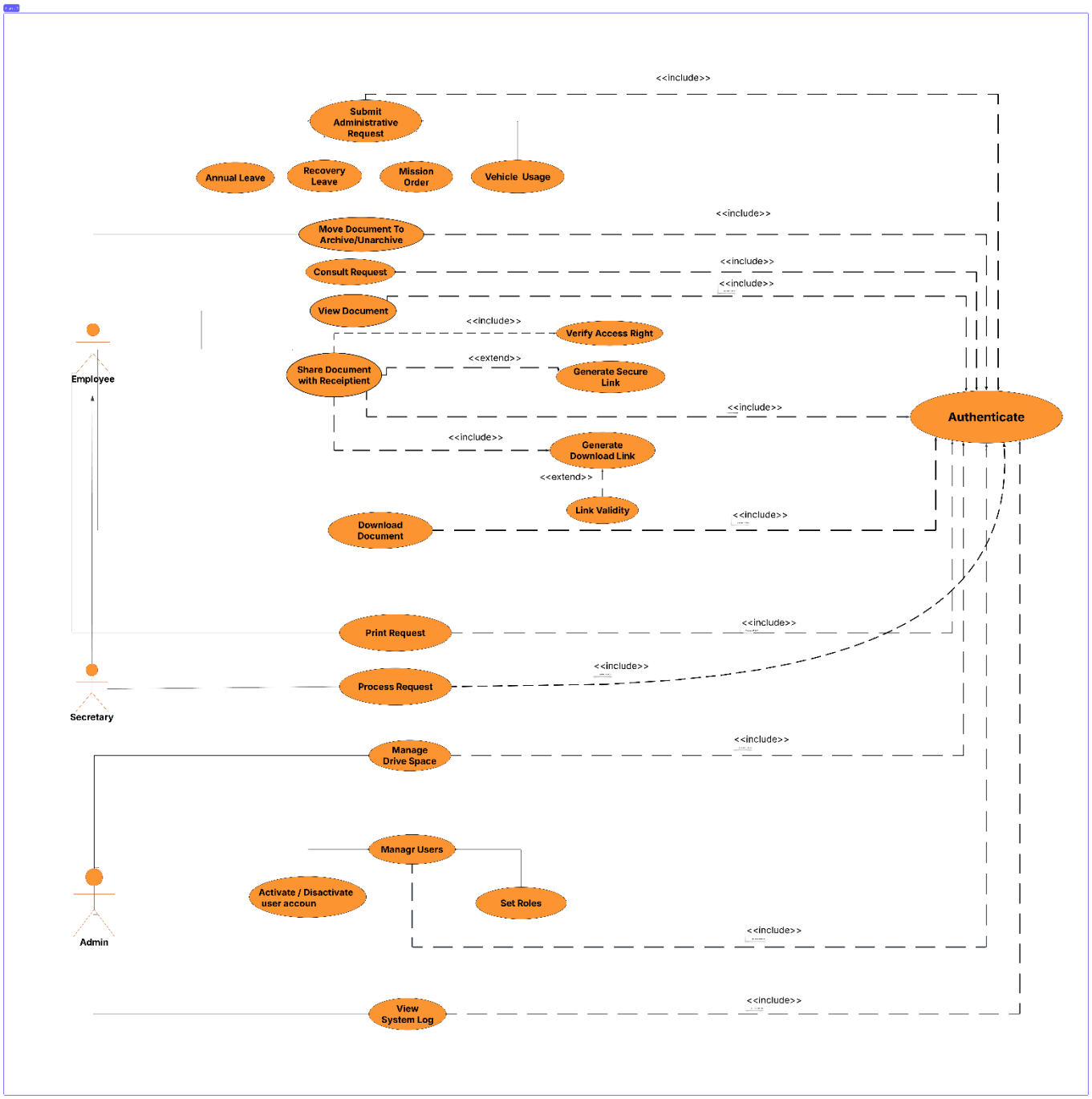


Figure 2.2: Global use case Diagram

3.2. Textual description of use case:

• Textual description of use case “Authenticate”:

	Description
Name	Authenticate
Actor	-primary actor: user: The person attempting to access the system. -Secondary actor: LDAP Server: the external authority that validates credentials.
Goals	To secure the login process of a user and authorizing access to the system’s Dashboard and secured features.
Pre-conditions	-The login interface is loaded and accessible - The LDAP server connection is active and responsive.
Textual Description	This use case covers how the system processes login by which user provides credentials to the system. The system delegates the validation to an external LDAP server and handles the session based on the server’s response.
Nominal scenario	1- The user enters their credentials into the Login form. 2- The system sends a” Request Verification” message to the LDAP Server. 3- The LDAP Server validates credentials and transmits a success response, the system authorized the session and grants the user access. 4- The system automatically redirects the client to the Dashboard.
Alternative scenario	Authentication Failed: the LDAP server sends a negative “Answer” (verification failed) or the system generates and displays an Error Message to the user or the system redirects the user back to the Login Form to re-enter credentials.
Post conditions	-Success: The system initializes a secure session and executes a redirection, routing the authenticated user to the personal Dashboard. -Failure: The LDAP response is negative, the authentication request is terminated and access is denied.

table 2.3: Textual description of use case”Authenticate”

• Textual description of use case "Submit Request":

Description	
Name	Submit Request
Actor	-primary actor: Employee. -Secondary actor: Secretary. -system component: Database.
Goals	To record an employee's administrative request in the database and notify the secretary for processing.
Pre-conditions	-The employee is successfully authenticated and the system is ready to accept data.
Textual Description	The employee submits a completed administrative form through the system, The system then manages the data storage, creates an audit log in the database, and alerts the secretary that a new request is pending.
Nominal scenario	1- Employee selects and fills out the request form. 2- System saves the request data to the database. 3- Database confirms the request is saved. 4- System logs the submission event in the database. 5- Database confirms the log is saved. 6- System sends a notification to the secretary. 7- The system notifies the employee of a successful submission.
Alternative scenario	Database Error: If the Database fails to save request or the log, the system displays a failure message to the employee and the process stops. Form Validation Error: The employee misses or incorrectly completes a required field, the system highlights the error and requests a correction
Post conditions	-Success: Request is saved and the secretary and the employee are notified. -Failure: No data is stored, the system displays an error message.

Table2.4: Textual description of use case "submit request"

• Textual description of use case "Share Documents":

Description	
Name	Share Documents
Actor	User (Employee, Secretary, Admin)
Goals	To allow a user to share a document with one or more users on the local server file system.
Pre-conditions	- The user is authenticated. - The document exists in the system.
Textual Description	This use case allows a user to select a document and share it with one or more recipients by updating access permissions.
Nominal scenario	The user selects a document. The user clicks the "Share" option. The system retrieves the document information from next cloud. The system displays the sharing options to the user. The user picks one or more recipients. The system updates the document access permissions in next cloud. The system validates the operation. The system displays a success message to the user.
Alternative scenario	- If no recipient is picked the system shows an error message. - If the document does not exist the whole operation is cancelled. - If cloud fails to update access permissions the system displays an error message.
Post conditions	The document is shared an accessible to the selected recipients.

Table 2.5: Textual description of use case "share document"

• Textual description of use case “Archive Documents”:

	Description
Name	Archive Documents
Actor	User (Employee, Secretary, Admin)
Goals	To allow a user to store a document in the archive for future access.
Pre-conditions	-The user is authenticated. -The document exists in the system.
Textual Description	This use case allows a user to move a document to an archive space controlled by the storage system while respecting storage quota.
Nominal scenario	1- The user chooses a document. 2- The user choses the “Archive” option. 3- The system retrieves the document information. 4- The system checks the available storage quota. 5- The system delivers the archive request to the storage service. 6- The document is displaced to the archive. 7- The system validate the operation.
Alternative scenario	-If the document is not available the whole operation is cancelled. -The system will show an error message if the storage quota is exceeded. -If something goes wrong with archiving, the system will display an error message.
Post conditions	The document is stored in the archive and can be find at a later time.

Table 2.6: Textual description of use case ”archive document”

4. sequence diagrams:

4.1. Global sequence diagram:

The global sequence diagram synthesizes all major interaction flows: authentication, request submission, notification, document sharing, archiving, and printing, within a single temporal representation. Its primary analytical value lies in establishing cross-cutting dependencies: authentication is a mandatory guard enclosing all other interactions, and the Employee and Secretary lifelines run in parallel, reflecting the asynchronous, decoupled nature of the notification model. The diagram answers the question of temporal ordering across the whole system, complementing the individual sequence diagrams which specify the internal flow of each use case.

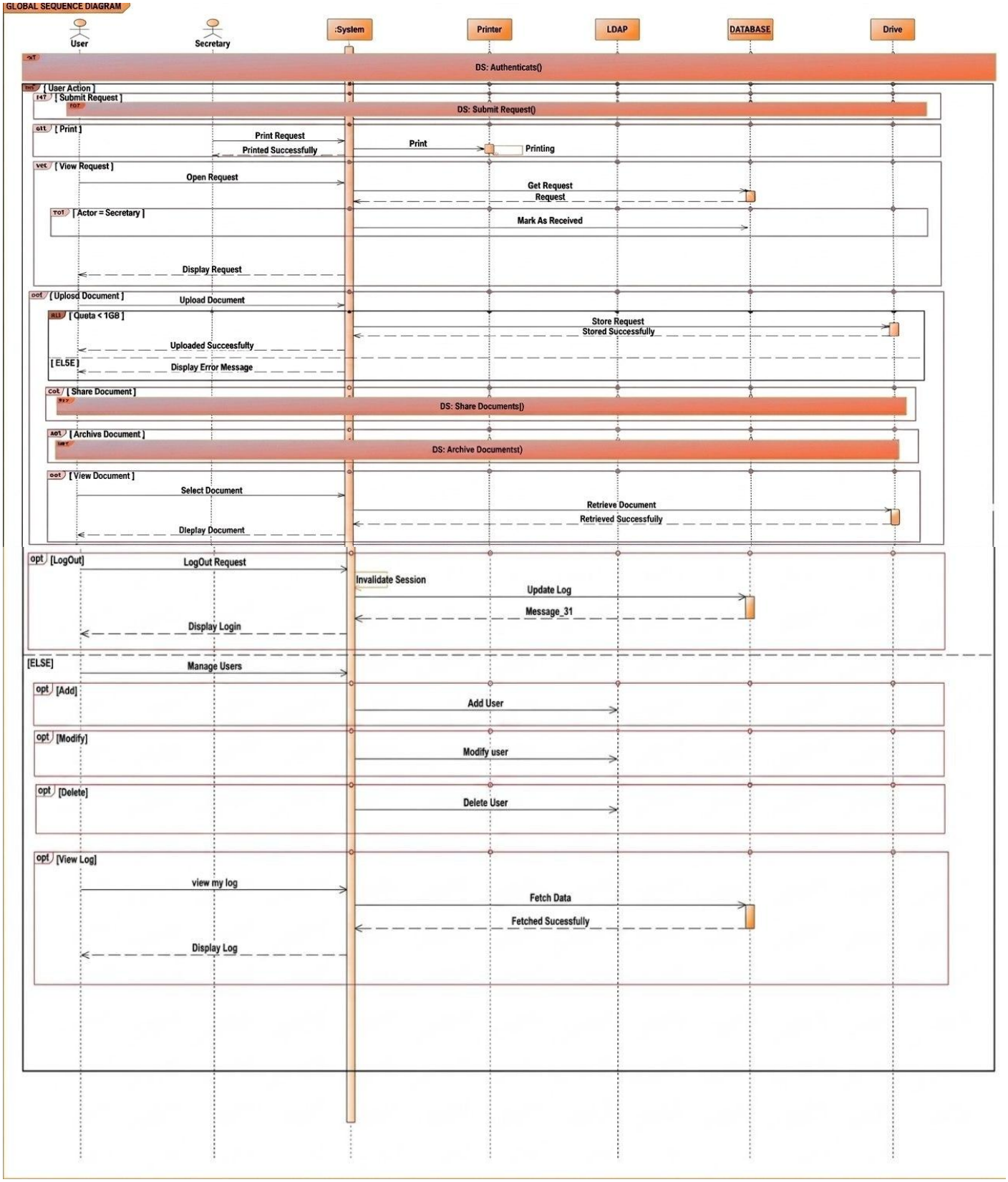


Figure 2.3: Global sequence diagram

4.2. sequence diagram of the use case “authenticate “:

The sequence diagram of the use case «Authenticate» illustrates the user authentication process through an LDAP-based mechanism where the local system acts as a security intermediary to enforce centralized identity management. After the user enters credentials, the system forwards them to the LDAP server to verify whether the user exists with an active account in the Active Directory. If authentication succeeds, access is granted, the login event is stored in the database for auditing purposes, and permissions are applied according to a Role-Based Access Control (RBAC) model. Otherwise, access is denied and an error message is displayed. This architecture ensures centralized authentication and separation between identity verification and data storage, although failed login attempts are not logged.

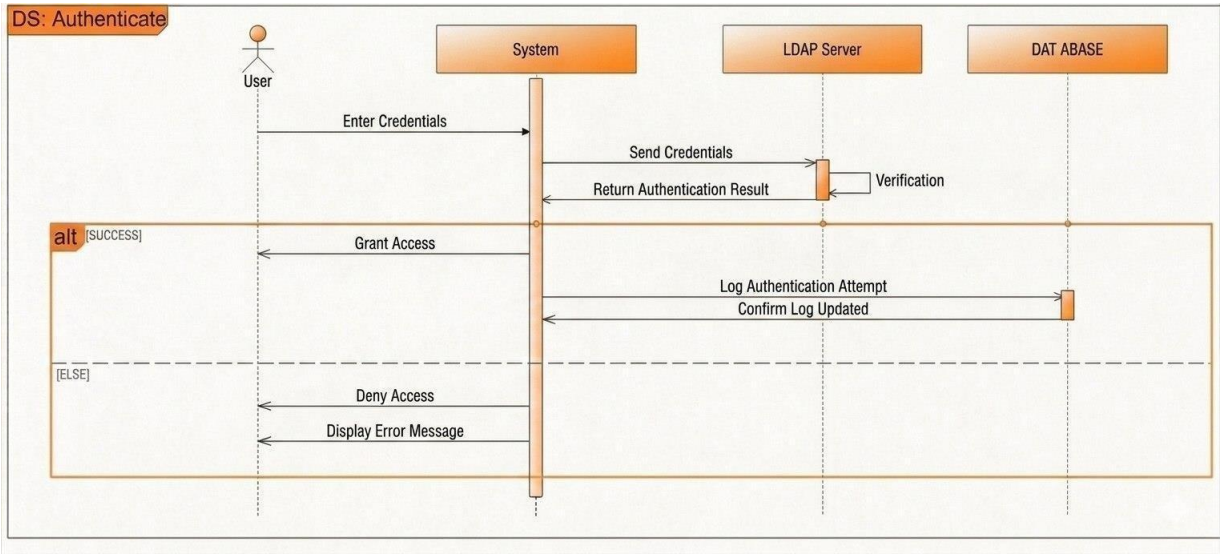


Figure 2.4: ‘’Authenticate’’ sequence diagram

4.3. sequence diagram of the use case “submit request “:

The sequence diagram for «Submit Request» models a secure administrative workflow gated by the “DS: Authenticate” fragment. Once an Employee submits a form, the System instantiates a Request object, logs the audit event in the Database, and notifies the Secretary. Any subsequent removal attempt is managed by a guarded deletion block (opt): if database checks reveal the request is already processed, the action is blocked; otherwise (ELSE), the system executes a <<destroy>> command, updates the database, and confirms the deletion to the employee.

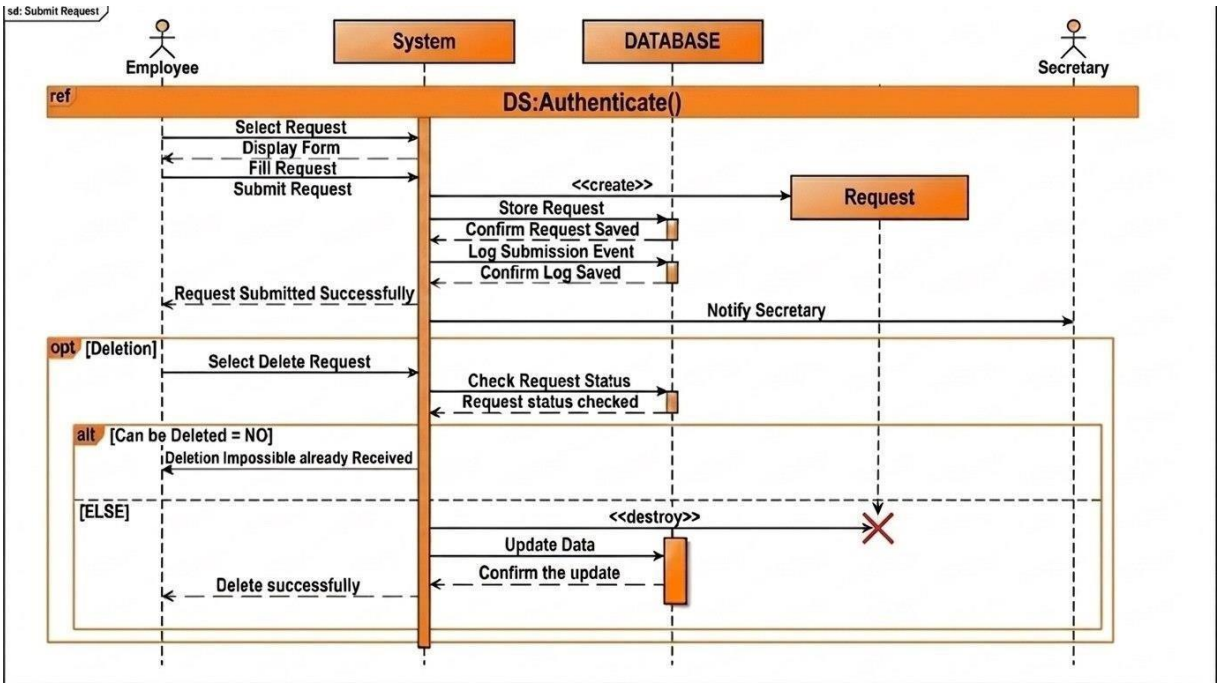


Figure 2.5:'' Submit Request '' sequence diagram

4.4. sequence diagram of the use case "share documents":

The sequence diagram for the use case «Share Documents» models a secure access delegation workflow gated by the "DS: Authenticate" reference fragment. Once authenticated, the User selects a document and requests sharing, prompting the System to fetch metadata from the DATABASE and return a Display Share Options response. Upon select Recipients the system transmits a "Grant Access to Recipients" and an opt fragment [secure link] that allows the user to generate a secure link and select the time limit, the database returns a Confirm Sharing Success verification, driving the system to deliver the final Notify User "Document Shared" acknowledgement to the User.

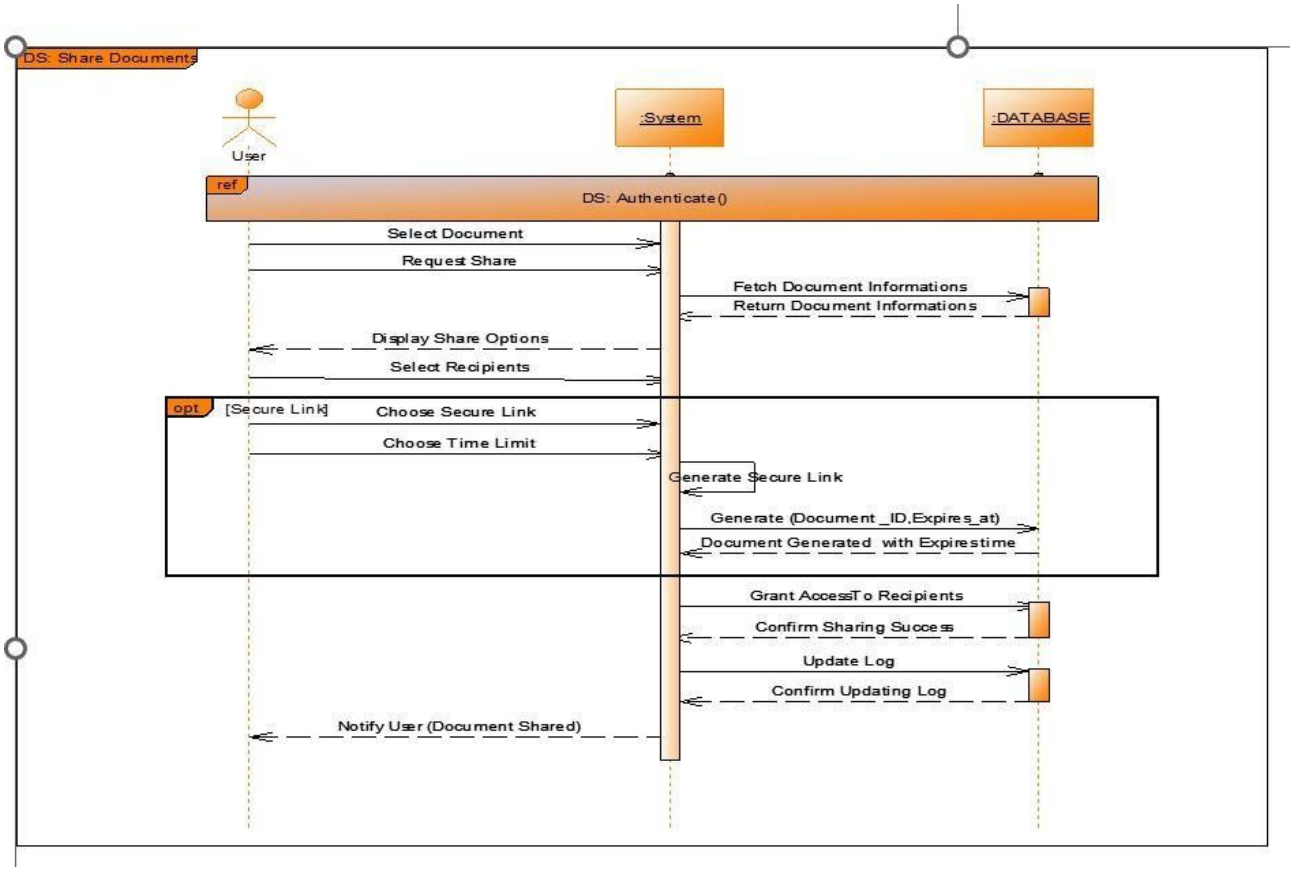


Figure 2.6:'' Share Documents '' sequence diagram

4.5. sequence diagram of the use case "archive documents":

The sequence diagram for the «Archive Documents» use case models a secure archival workflow initiated through the" DS: Authenticate" reference fragment to ensure authorized access. After authentication, the User selects a document and submits an archive request to the System, which retrieves the corresponding document data from the DATABASE. The System then executes the archival operation, updates the log, and receives a success confirmation from the database layer before notifying the User that the document has been successfully archived. Any subsequent unarchive operation is managed by a "opt" fragment, the system executes the operation, update the log and confirms.

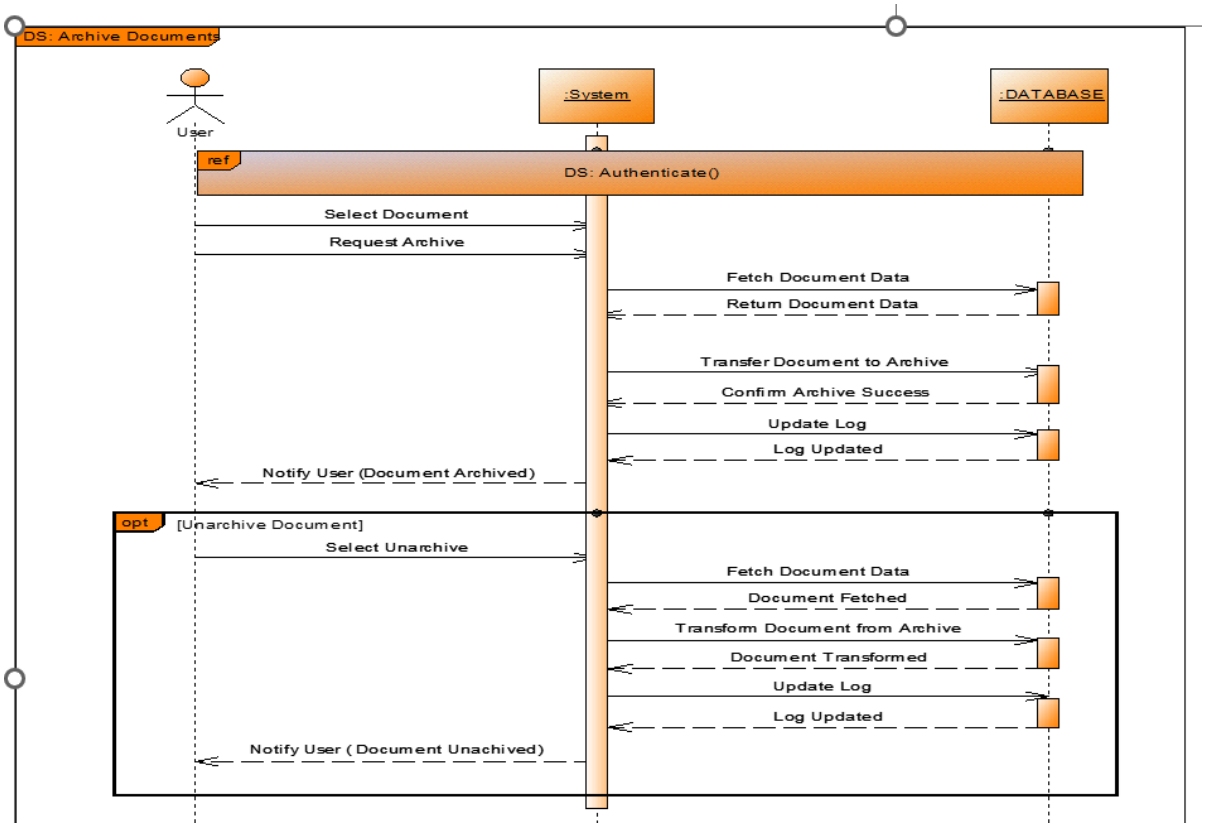


Figure 2.7: '' Archive Documents '' sequence diagram

Conclusion:

This chapter established the system's core requirements, through Use Case and Sequence diagrams, setting the ground for the class diagram and database design in the next chapter.

Chapter 03:

System Design

introduction:

In this chapter, we converted the project requirements into a technical design. We will define the 3-tier architecture, create the UML Class Diagrams, and setup the database models (MCD and MLD) to make sure the application is secured and organized before we start the development.

1. System architecture:

1.1. General architecture (3-Tier Architecture):

The system is based on a three-tier architecture and integrates an authentication service using Active Directory to manage user access and security.

- **Presentation Tier:** Acting as the primary user interface and communication layer, this top-level tier manages user interaction by displaying and collecting information. It can be implemented via web browsers (utilizing React), desktop applications, or graphical user interfaces (GUIs).

- **Application Tier:** Serving as the core logic or middle tier, this layer processes data collected from the presentation tier by enforcing specific business rules and logic. It governs data modification within the storage layer and typically relies on languages PHP (Laravel) to interact with the data tier via API calls.

- **Data Tier:** This back-end layer functions as the persistence and management system for application data. It can be implemented using relational database management systems (such as MySQL).

- In addition, Active Directory is used as an external service for authentication. It verifies user credentials and improves security by avoiding the storage of sensitive information in the database, and it is integrated using LDAP for user authentication.

- **Document management operations** (uploading, sharing, archiving…etc) are handled using a Drive.

1.2. Architecture Diagram:

The diagram formalizes the separation of concerns across Presentation (React), Application (Laravel/PHP with document management services), and Data (MySQL) tiers, interconnected through HTTP/REST and PDO/MySQL communication interfaces respectively. Active Directory operates as an external federated identity service associated with the application tier and accessed through LDAP-based authentication flows.

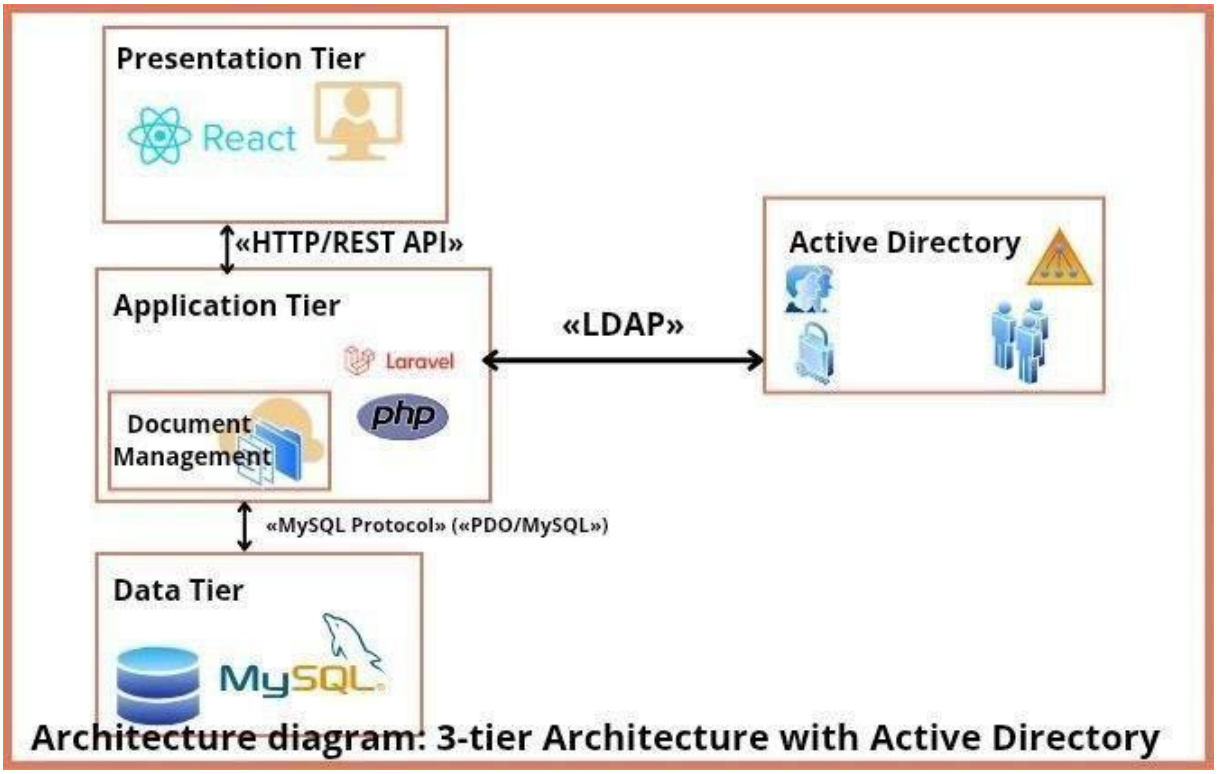


Figure 3.8: Architecture diagram

2. Static Modelling (UML):

2.1. Class Diagram:

The class diagram formalizes the static structural model of the system through nine domain entities: User, Request, Document, Archive, Notification, ShareLink, DocumentShare, Group and Log. The User class encapsulates identity and authorization attributes, including a Role enumeration: Employee, Secretary, and Admin, alongside storage consumption tracking. The Request class models administrative requests via a Type enumeration (Annual Leave, Recovery Leave, Mission Order, Vehicle), with lifecycle operations Submit (), Delete(),canBeDeleted(), and MarkAsReceived(). The Document class acts as the core content entity, supporting upload, download, archival, sharing, and secure link generation.
Archive records the relocation of documents with restoration capability, while ShareLink implements token-based, time-bounded access delegation. Log provides immutable audit traceability per user action.



Figure3.9: Class diagram

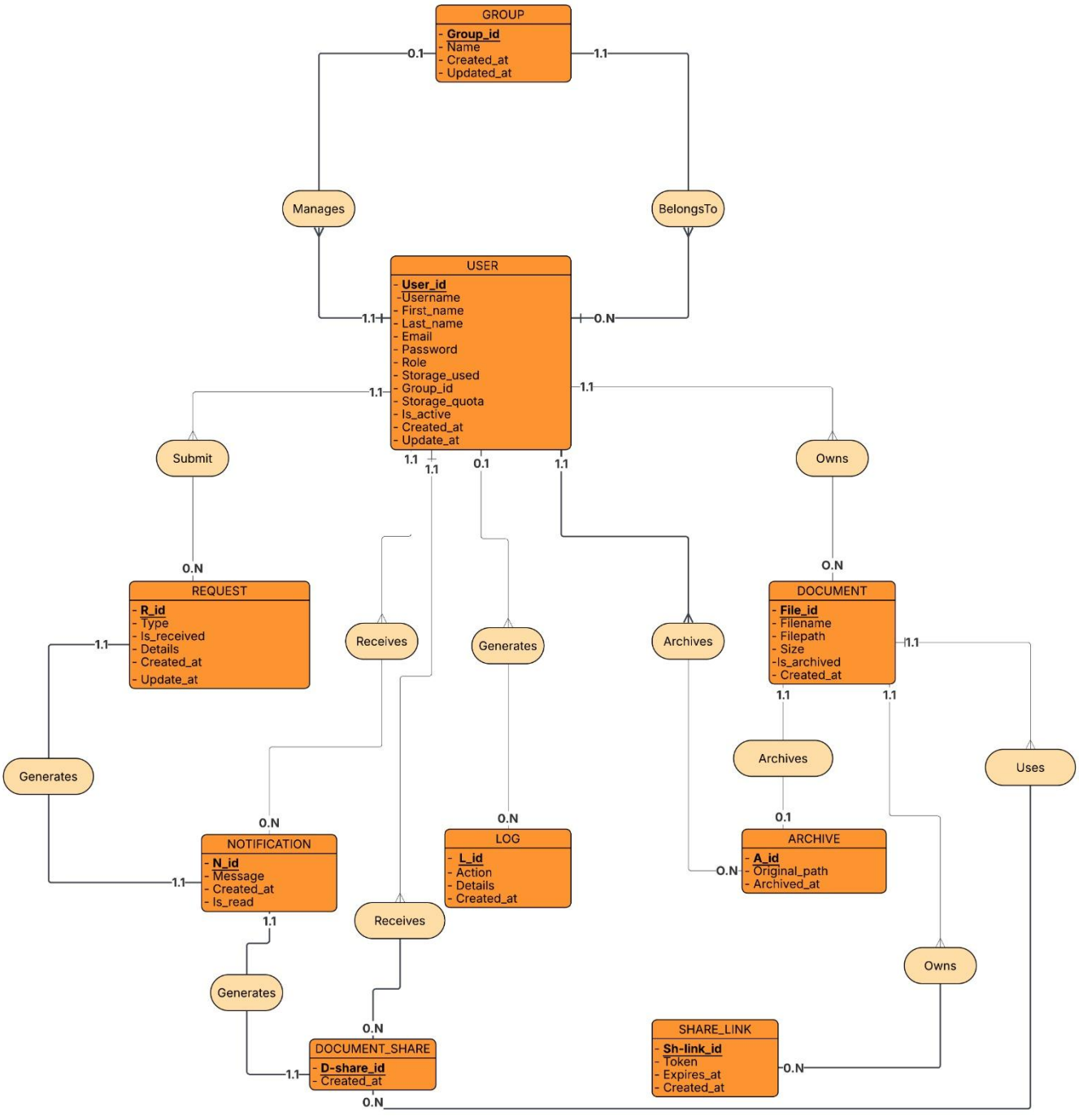
3. Data Base Design:

This section introduces the data model of the GCB application using the Merise methodology: from the Conceptual Data Model (MCD) to the Logical Data Model (MLD) and the physical MySQL schema.

The Merise methodology was chosen for the database design phase due to its structured, two-level modeling approach (conceptual then logical), which is well-suited to complex enterprise information systems. It allows a clear separation between the Conceptual Data Model (MCD) and its technical relational implementation (MLD), aligning seamlessly with established engineering practices and data architecture standards in the Algerian corporate ecosystem.

3.1. Conceptual Data Model (MCD):

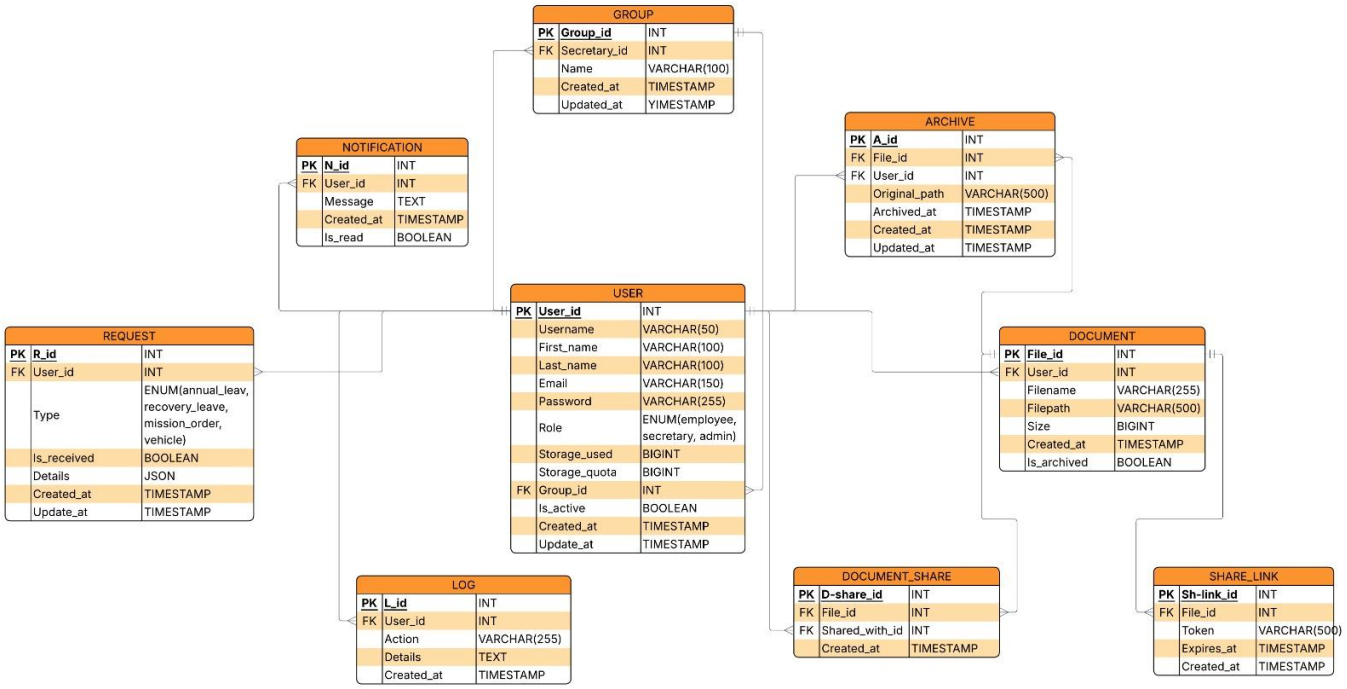
The MCD represents nine entities with GROUP at the top, linked to USER via Manages (1,1) -(0,1) and BelongsTo (1,1) -(0,N). USER is the central entity, participating in Submit with REQUEST (1,1) -(0,N), Receives with NOTIFICATION (1,1) -(0,N), Generates with LOG (0,1) -(0,N), Archives with ARCHIVE (1,1) -(0,N), and Owns with DOCUMENT (1,1) -(0,N). REQUEST triggers NOTIFICATION via Generates (1,1)-(1,1), while USER connects to DOCUMENT_SHARE via Receives (1,1) -(0, N). DOCUMENT links to ARCHIVE via Archives (1,1) -(0,1) and to DOCUMENT_SHARE via Uses (1,1) -(0,N), with SHARE_LINK through Owns (1,1) -(0,N). DOCUMENT_SHARE resolves the sharing association, linked back to NOTIFICATION via Generates (1,1) -(1,1).



3.10. Conceptual Data Model (MCD)

3.2. Logical Data Model (MLD):

The MLD formalizes key constraints through PK/FK annotations. USER is the pivot relation linked to REQUEST, NOTIFICATION, LOG, DOCUMENT, and ARCHIVE via User_id. GROUP connects to USER through Secretary_id FK.ARCHIVE holds dual FK references (File_id and User_id) for ownership traceability. REQUEST uses Type ENUM and Details JSON for flexible payload structures. DOCUMENT_SHARE resolves many-to-many sharing via File_id and Shared_with_id. SHARE_LINK depends solely on DOCUMENT via File_id, enforcing time-bounded access through Token VARCHAR (500) and Expires_at. All relations carry Created_at; REQUEST and USER additionally hold Update_at reflecting their mutable lifecycles.



3.11. Logical Data Model (MLD)

3.3. Data Base Schema:

The schema defines nine relations with USER as the central hub, propagating User_id into REQUEST, NOTIFICATION, LOG, DOCUMENT, and ARCHIVE. USER links to GROUP via Group_id, while GROUP back-references USER through Secretary_id to designate an administrative secretary. DOCUMENT acts as a secondary hub, referenced by ARCHIVE, SHARE_LINK, and DOCUMENT_SHARE via File_id. ARCHIVE additionally holds User_id NN, recording which user triggered the archival. DOCUMENT_SHARE adds a second foreign key Shared_with_id to resolve user-to-document sharing. Notable design choices include an enumerated Role and request Type, a JSON column in REQUEST for flexible payload structures, bigint for storage quota precision, and SHARE_LINK.Token varchar (500) paired with Expires_at for time-bounded access delegation.

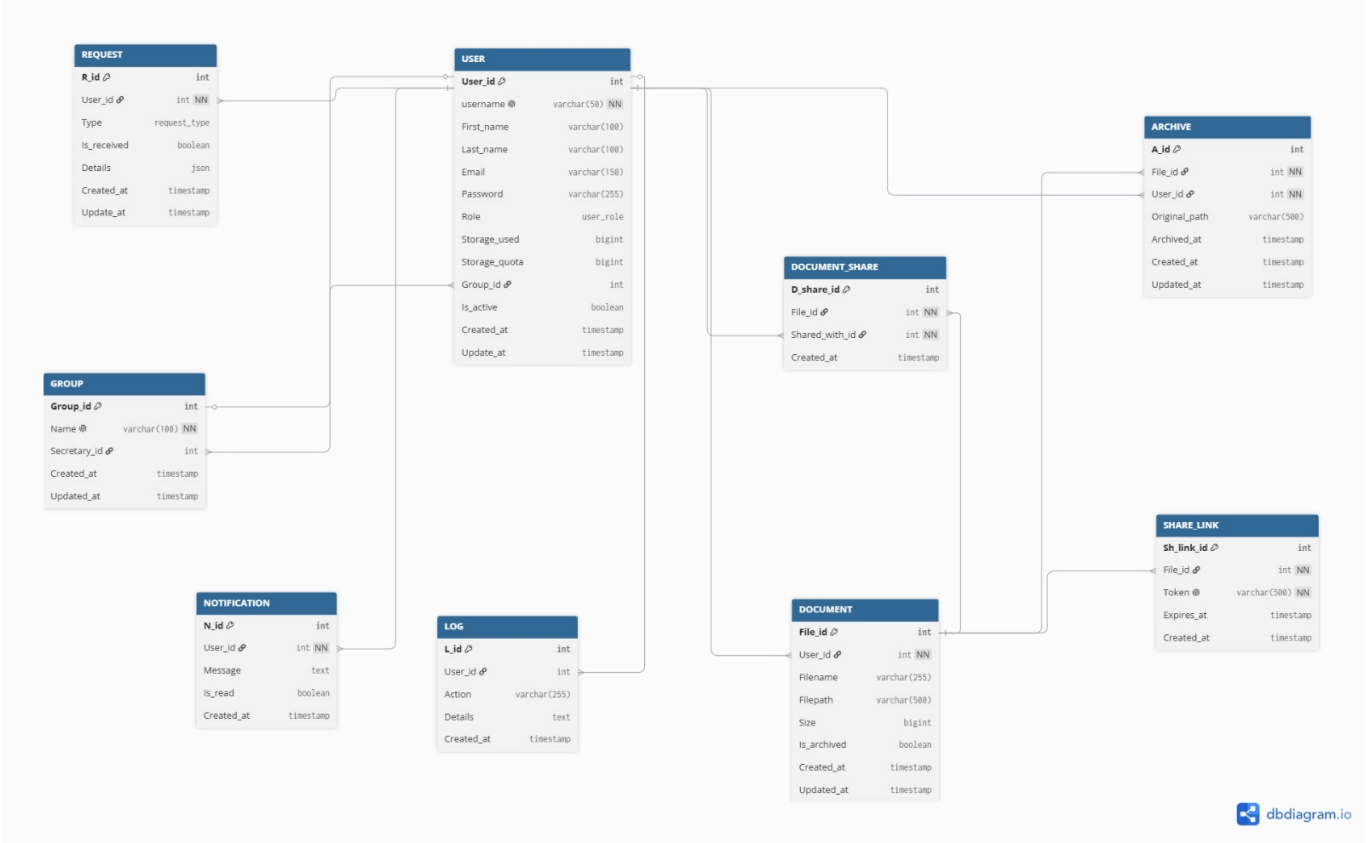


Figure3.12: Data Base Schema

conclusion:

This chapter provided the technical plan for the application, by defining the system architecture and the database schema, we ensured that the system is prepared to handle key requirements such as LDAP authentication and storage limitations. With the design phase completed, the project is now ready to move forward to implementation, followed by deployment diagram and the presentation of the application.

Chapter 04:

System Realization and Implementation

introduction:

This chapter presents the technical implementation of the GCB administrative management system. It covers the development environment, the technology stack used, the implementation of the core modules (authentication and document management), and a visual presentation of the final application.

1. Deployment Diagram:

The diagram depicts a four-node physical topology in which a React client communicates with a Laravel application server through “HTTP based REST API”. The application server centralizes RBAC, request management, and quota-enforced personal storage services, while delegating identity resolution to Active Directory via LDAP. Persistence is provided by a dedicated MySQL node accessed through PDO/MySQL connectivity and logically partitioned into user, request, and document data domains.

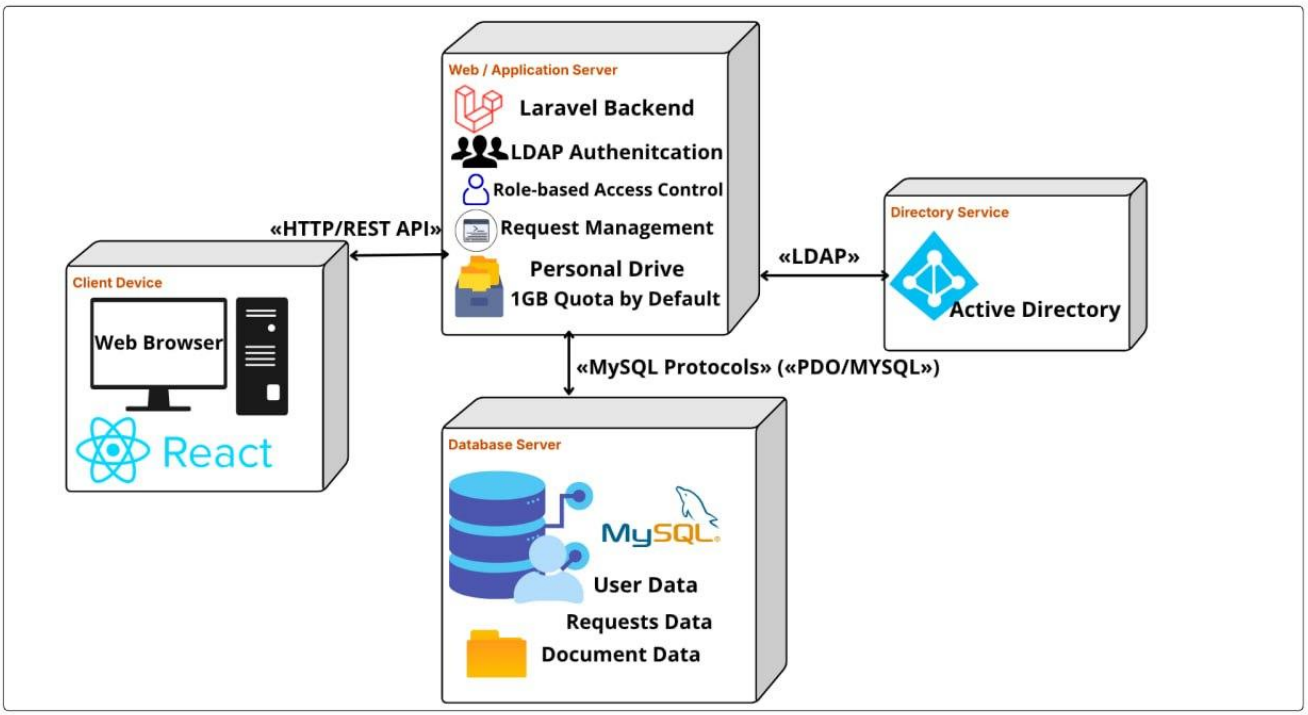


Figure 4.13: Deployment diagram

2. Development Environment:

2.1. Hardware environment:

- **Machine 1:**

Category	Details
Machine Name	DELL latitude E5450
Processor	Intel (R) Core (TM) i5-5300U CPU@ 2.30GHz
RAM memory	8 GB
System Type	64-bit operating system, x64- based processor
Operating System	Windows 10 pro

Table 4.7: Specification machine 1

- **Machine 2:**

Category	Details
Machine Name	DELL latitude 3420
Processor	11th Gen Intel (R) Core (TM) i5-1135G7 CPU@ 2.40GHz
RAM memory	16 GB
System Type	64-bit operating system, x64- based processor
Operating System	Windows 11

Table 4.8: Specification machine 2

- **Machine 3:**

Category	Details
Machine Name	HP
Processor	11th Gen Intel (R) Core (TM) i5-1145G7 CPU@ 2.60GHz
RAM memory	16 GB
System Type	64-bit operating system, x64- based processor
Operating System	Windows 11

Table 4.9: Specification machine 3

2.2. Software environment:

Tool	Justification	Version	Tool Logo
Visual Studio Code	A lightweight source code editor used for development.	1.111.0	
Laragon	Designed to increase productivity and simplify web development, and serves as a modern, actively maintained, and feature-rich local development environment.	v8.6.1	
Windows Server2022 (Hyper-v)	A virtualization technology for creating virtual machines. Advanced multi-layer security, and a flexible application platform.	Standard	
Power Designer	Efficiently visualize, analyze, and manage metadata, used to draw sequence diagrams.	v15.1	
Chrome	Used for testing, debugging, and accessing the web application during development.	Web	
Draw.io	For building diagramming applications, used to draw architecture diagram.	Web	
Lucid chart	A diagramming tool used to create system and data models, used to draw MCD, MLD, class diagram, use case diagram and deployment diagram.	Web	
dbdiagram.io	Open-access tool that simplifies the generation of Entity-Relationship (ER) diagrams through a code-based interface, used to draw Database.	Web	
Bruno	An offline-first API client designed to facilitate efficient, local developer Workflows through plain-text collections.	3.3.0	
Composer	Functions as a dependency management utility for the PHP ecosystem, enabling developers to define project-specific library dependencies and automating their installation and maintenance lifecycle.	6.6.0.0	
Node.js(npm)	An open-source, cross-platform JavaScript runtime environment that enables the execution of JavaScript code. npm is an open-source package manager used to install and share packages.	V24.13.1	

Table 4.10: Software environment

2.3. Technology Stack:

Layer	Technology	Version	ification
Frontend	React.js (Vite)	@19.2.5 @8.0.10	Enables the construction of user interfaces through reusable and nestable components, which supports the development of interactive interfaces in a structured and efficient way. Provides a fast development environment with instant server startup and efficient Hot Module Replacement (HMR), allowing rapid reflection of code changes during development.
Backend	Laravel	@12.57.0	Was selected due to its built-in features such as authentication mechanisms, security protections, and the Eloquent ORM, which simplifies database operations and improves backend development efficiency.
ackend Language	PHP	@8.3.30	Widely recognized as a scripting language particularly suited for web development and is supported across most hosting environments.
Database	MySQL	@8.4.3	A relational database designed to efficiently store and manage structured data while ensuring reliable query processing.
Authentication	LDAP + Sanctum	Ldapv3	Integrated to enable centralized user authentication, allowing users to access the system using existing organizational credentials managed through directory services.
HTTP client	Axios	@1.16.0	An HTTP client to facilitate communication with APIs, offering features such as request and response interceptors, which help manage authentication tokens and request handling.
Apache	Apache HTTP server	@2.4.66	Was selected as the web server because it is a widely used, open-source HTTP server designed to efficiently host web applications with flexibility and reliability.

Table 4.11: Technology stack

3. Implementation of core module:

3.1. Authentication and directory integration:

The authentication system of the GCB application is built on two integrated layers. The first layer relies on Active Directory (AD), the directory service already deployed on the company's Windows Server. When an employee attempts to log in, the application connects to Active Directory via the LDAP protocol (Lightweight Directory Access Protocol) on port 389, and verifies the credentials against the existing company user base. This means no separate user registration system is needed every GCB employee can log in using their existing Windows credentials.

The second layer uses Laravel Sanctum for API token management. Upon successful AD verification, the system automatically creates the user's profile in the MySQL database if it does not already exist (using the first Or Create mechanism), assigns them the default role of employee, and generates a secure authentication token. This token is then sent with every subsequent request in the Authorization header, allowing the server to identify the user and their role without repeating the AD verification on each request.

This hybrid approach combines the security and centralized management of Active Directory with the flexibility of token-based API authentication, ensuring that user lifecycle management (hiring, departure, password changes) remains entirely under the control of the company's IT department.

3.2. Cloud Storage Integration:

The document management module provides each user with a personal storage space of 1 GB directly on the application server, without relying on any external cloud service. Files are stored in the server's file system under an organized directory structure **(`storage/app/public/document/{user id}/`)**, while their metadata (filename, path, size, owner) are recorded in the MySQL database. The system enforces storage quotas at the application level: before each upload, the server calculates the sum of the user's current storage_used value and the size of the new file. If the total exceeds 1073741824 bytes (1 GB), the upload is rejected with an appropriate error message. When a file is deleted, its size is automatically deducted from the user's quota. Beyond simple storage, the module supports document sharing between users (either by directly selecting recipients or by generating secure download links with configurable expiration periods). Documents can also be archived, creating a historical record of important files. This architecture keeps all company data within the internal network, which aligns with GCB's data confidentiality requirements and eliminates dependency on third-party cloud providers.

4. Presentation of the graphical interfaces:

Once the application is launched, the system displays an interface representing the login page of our application. Through this interface, users can use the system each according to their rights. We will provide a description of each window in relation to the different interfaces that constitute our application, depending on the user.

4.1. Authentication page:

The login interface of the GCB Portal provides secure Role-based access, Users authenticate using their Active Directory credentials. The interface supports bilingual navigation (French/English).

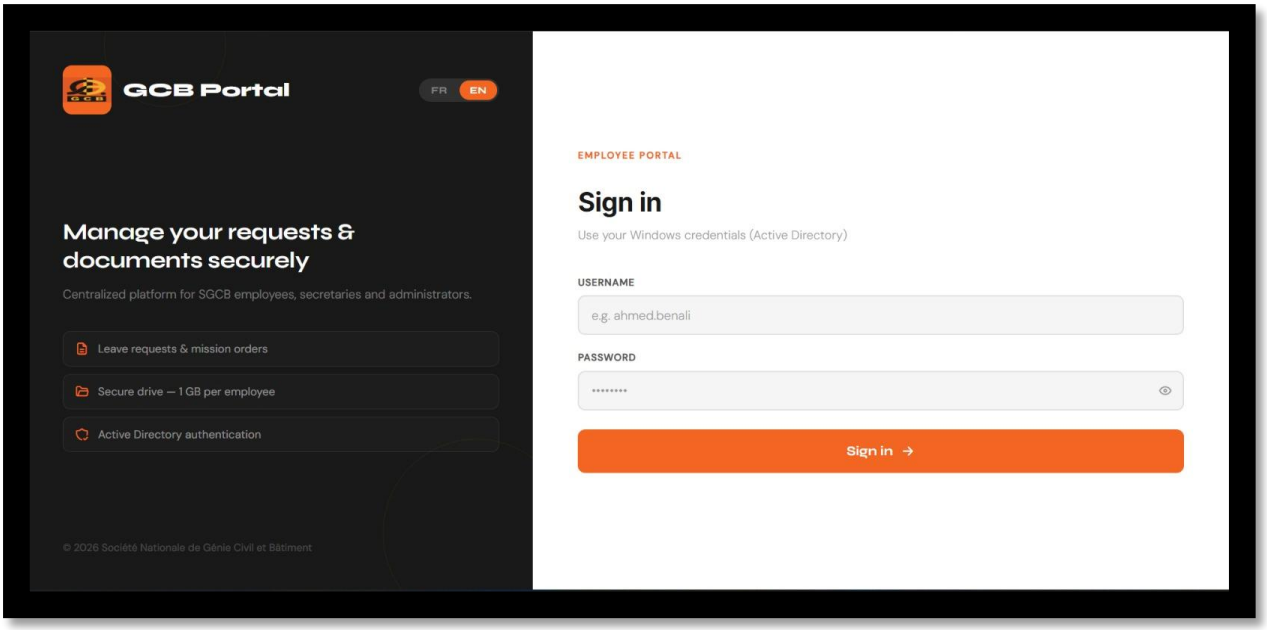


Figure 4.14: Login page

4.2. Employee space:

●Employee Dashboard:

The main dashboard provides employees with a consolidated overview of their activity within the portal, including the total number of submitted requests, pending items, and processed requests, alongside current drive storage consumption. Quick-access shortcuts facilitate the initiation of common request types (annual leave, mission order, vehicle request, recovery leave), while a recent requests table offers real-time tracking.

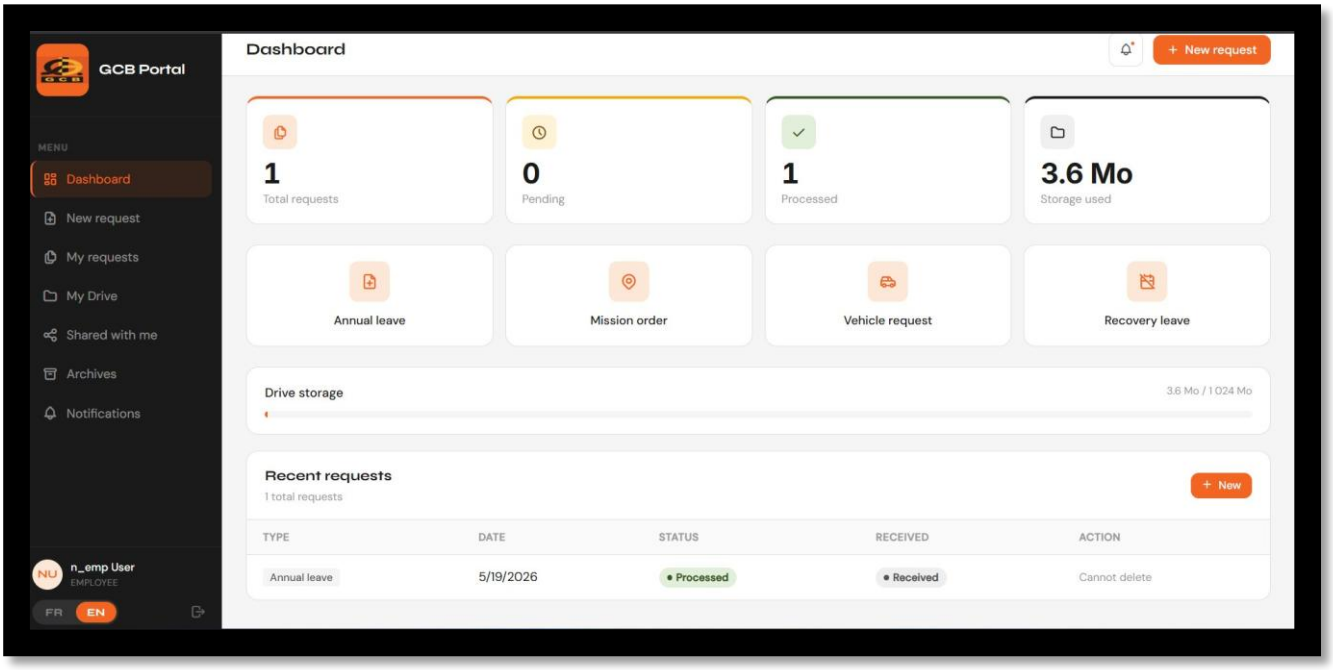


Figure4.15: Employee Dashboard

●Submit Request:

The new request interface presents employees with a structured selection of available administrative request categories: Vehicle Request, Leave Request (CP/CR/CS), and Mission Order.

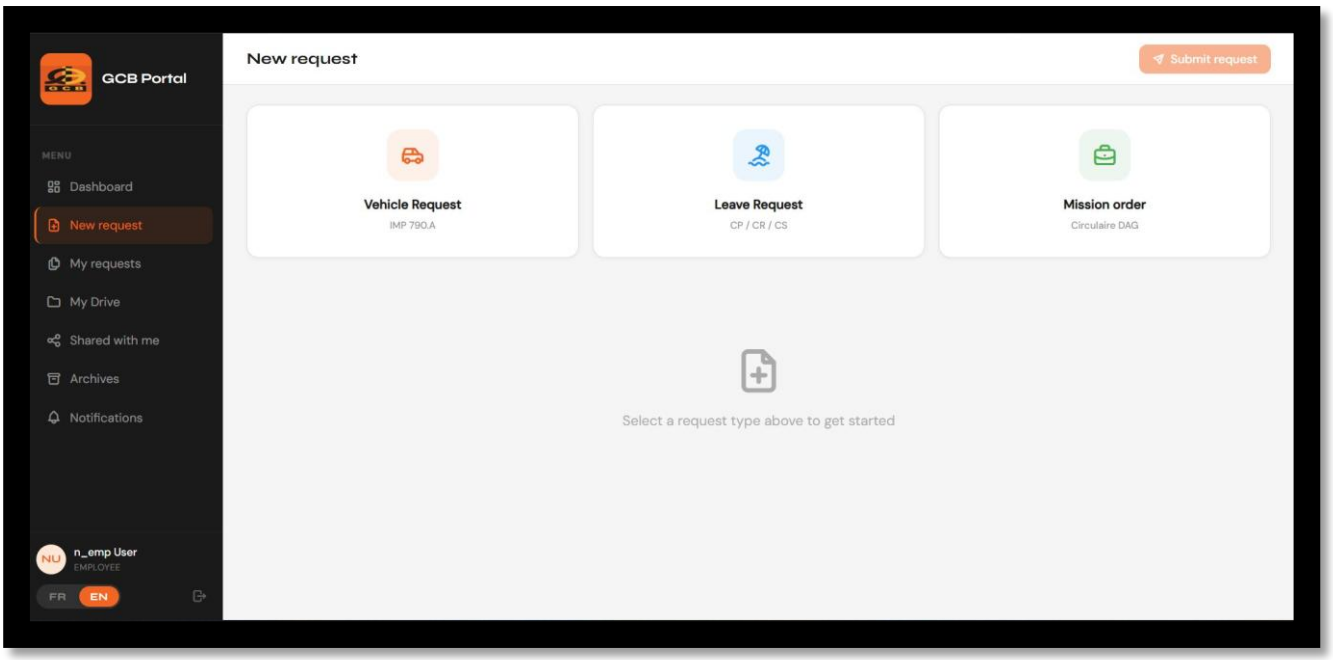


Figure4.16: New Request

The Leave Request form enables employees to submit formal leave applications, (CP: Annual Leave, CR: Recovery Leave, CS: Sick Leave). To ensure proper processing, all required fields must be fully completed before submission.

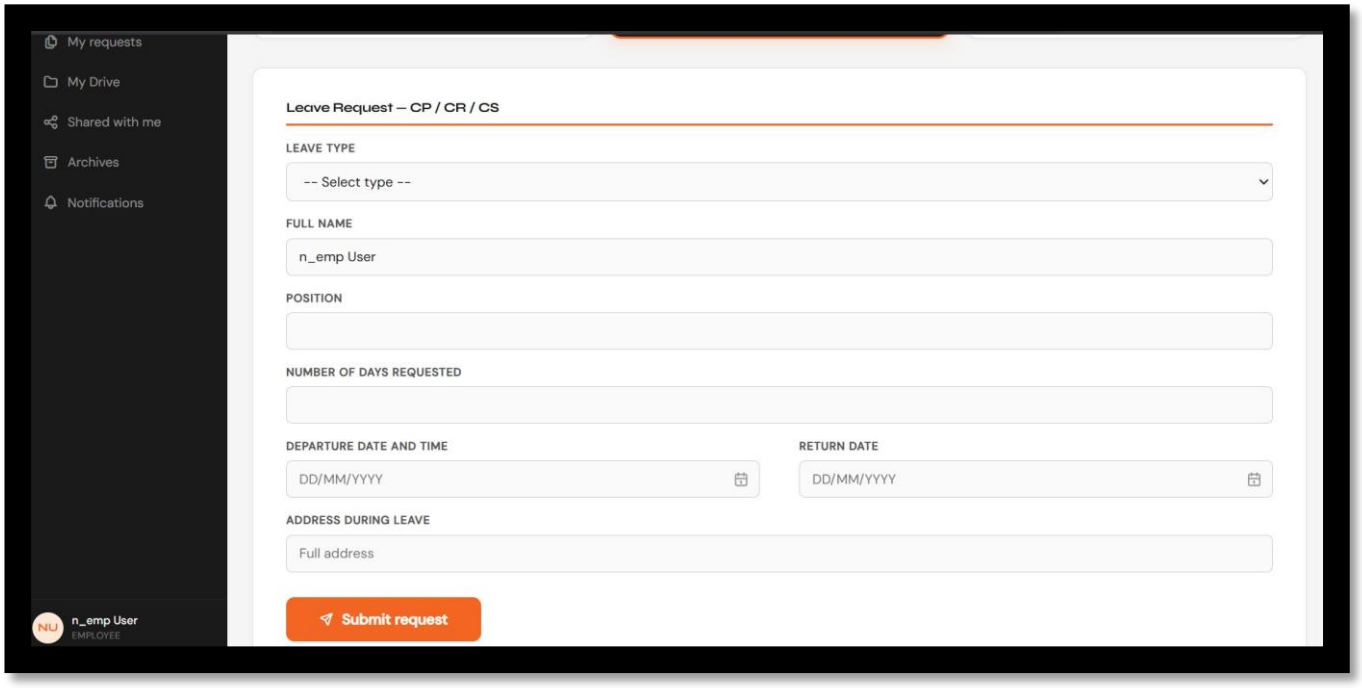


Figure4.17: Example of Request "Leave Request"

Upon selecting the Vehicle Request category, the system renders a structured data entry form. This standardised form ensures that all vehicle allocation requests adhere to the organisation's internal logistics and administrative requirements.

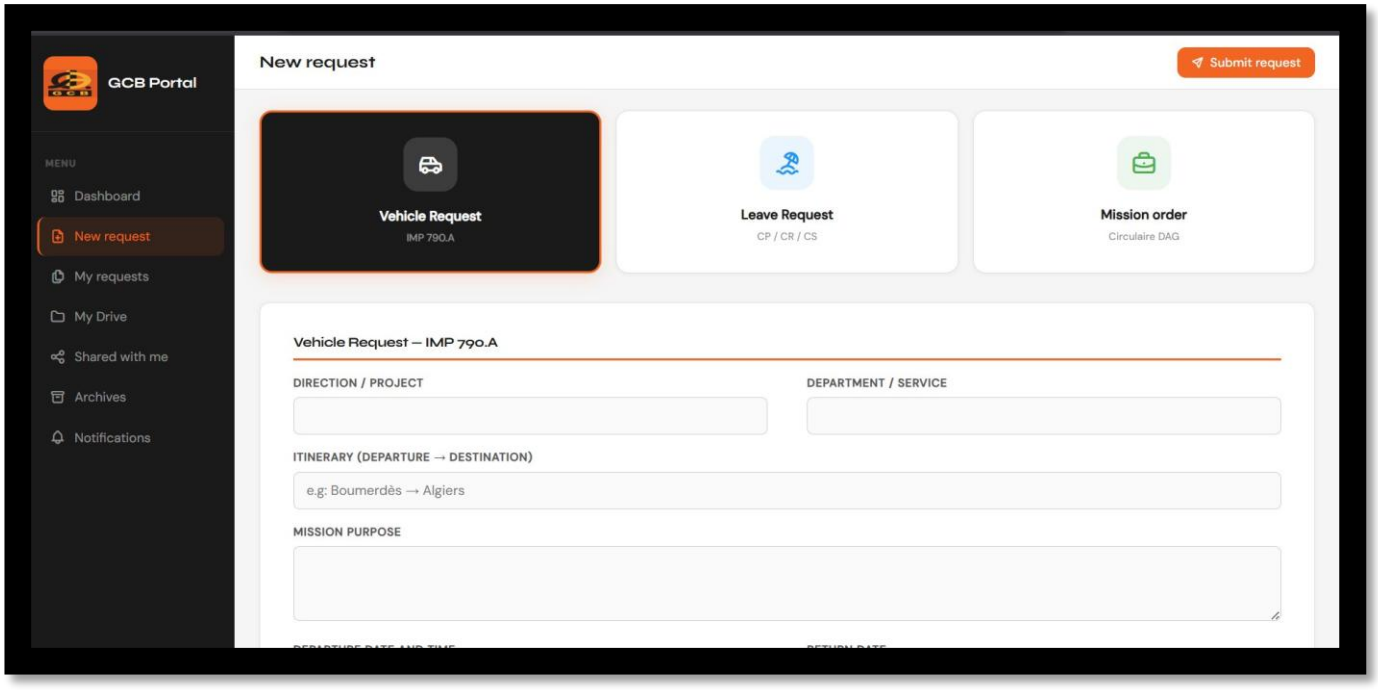


Figure4.18: Example of Request "Vehicle Request"

●My Request:

The My Requests module provides employees with a record of all administrative requests submitted through the portal. Each entry displays the request type, submission date, current processing status, and the corresponding action available to the user. Once a request has been received, the deletion action is disabled.

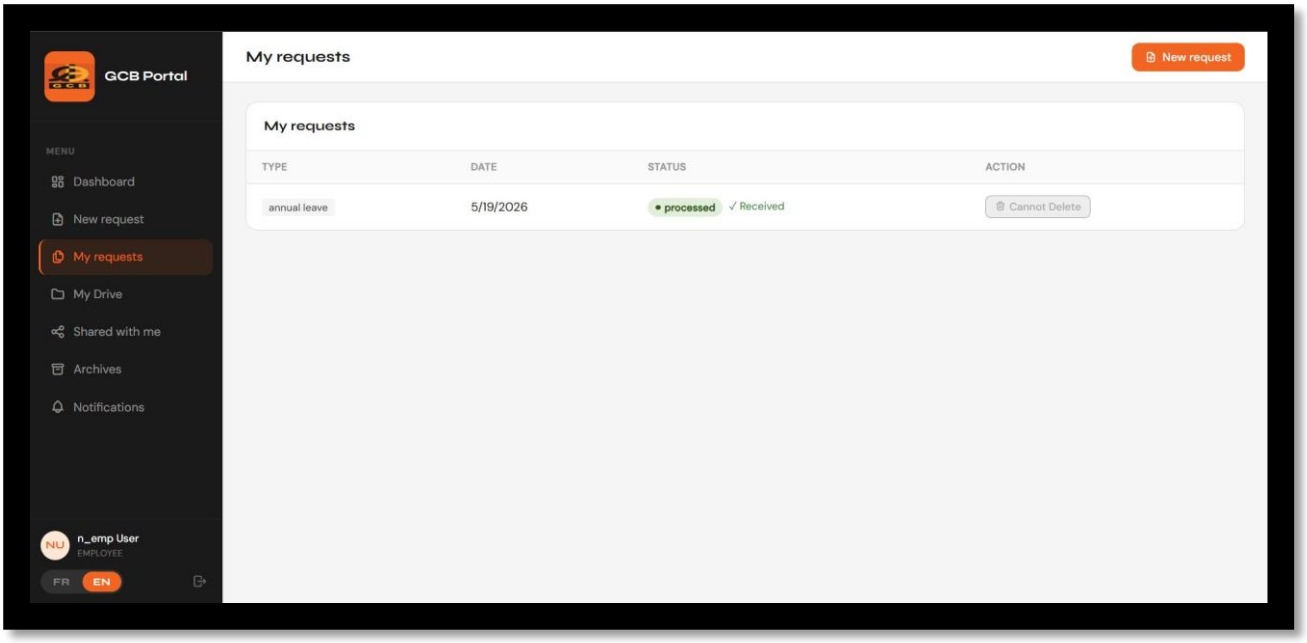


Figure4.19: History of Requests

●Drive Space:

The My Drive module provides employees with a personal storage space up to a 1 GB quota. The file list displays each item's name, size, and date. Users can comprehensively manage these files directly through integrated Download, Share, Archive, Print, and Delete actions.

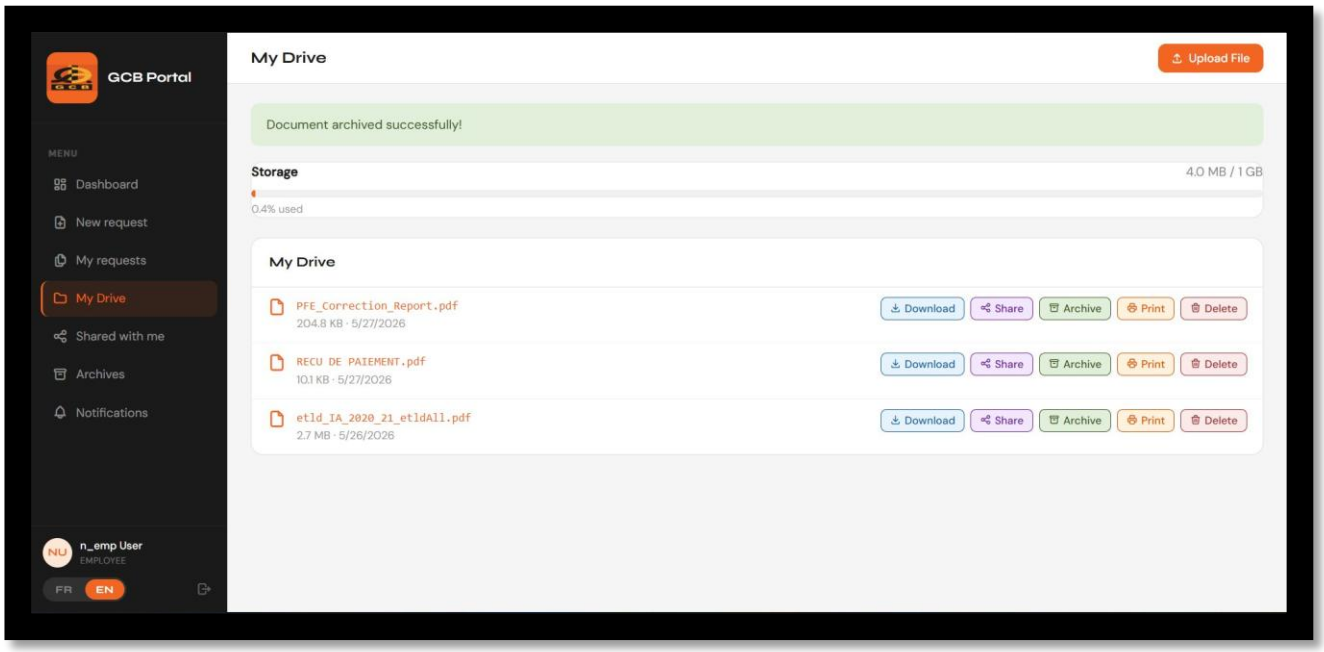


Figure4.20 : Drive Space

●Archive Documents :

The Archives section serves as a long-term repository for documents explicitly removed from the active drive. Each entry retains its original filename and archiving date to ensure full lifecycle traceability. Users can directly manage these files using the available Download, Print, Unarchive, or Delete actions.

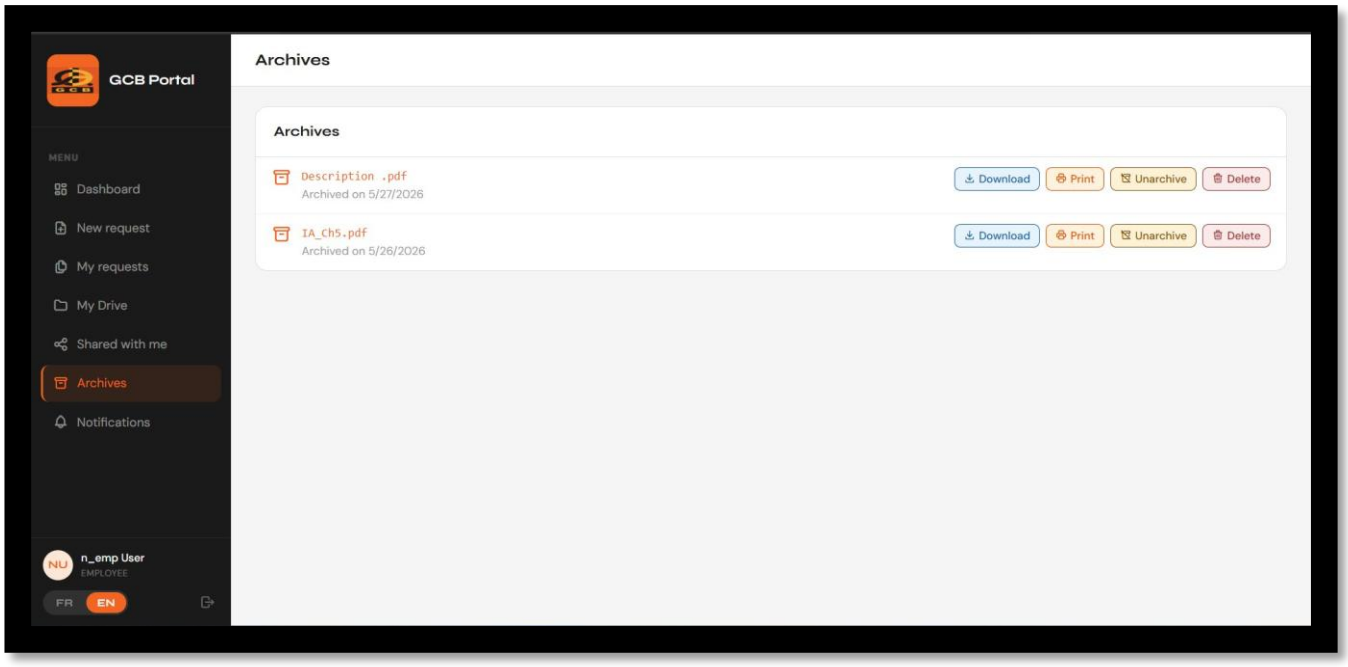


Figure4.21: Archive Documents

●Share with me:

The Shared with Me module aggregates documents that have been shared with the authenticated employee by other portal users. Each entry identifies the document name, the originating user, and the date of sharing, alongside a dedicated download button for quick access.

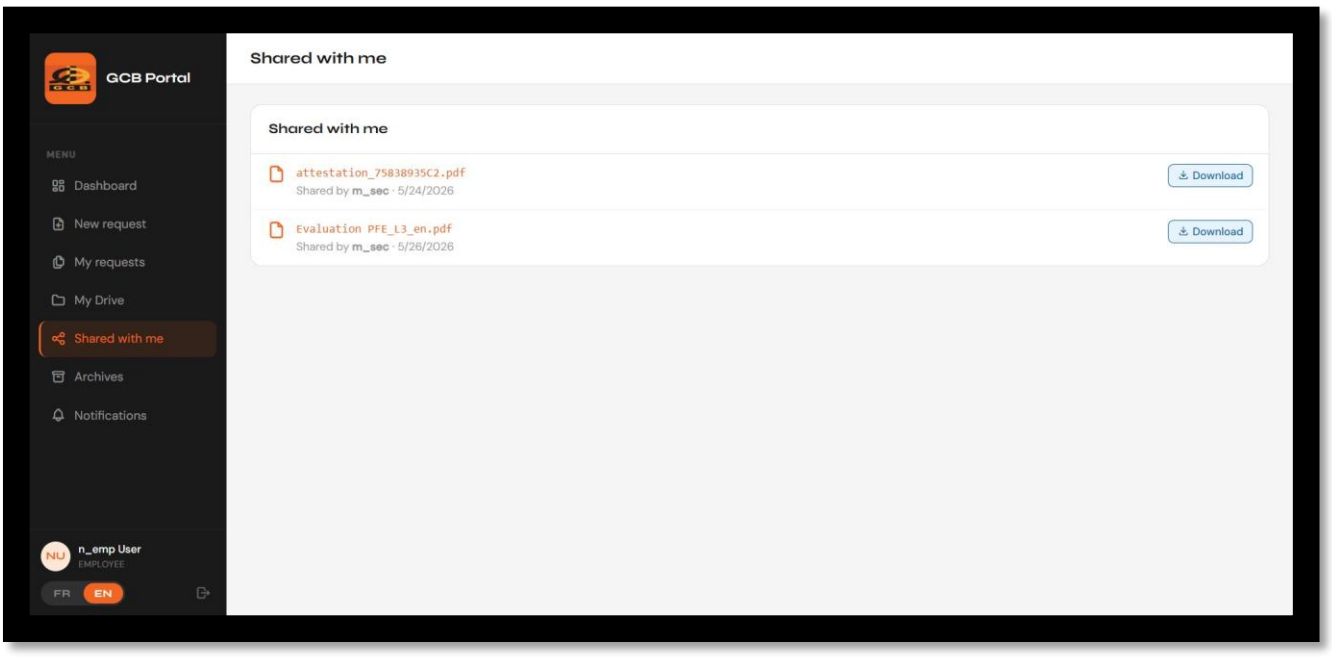


Figure4.22: Shared with me

●Share with people:

The document sharing module allows authenticated users to distribute files to selected colleagues which is identical for both secretaries and employees, it is through a searchable user directory, with configurable download permissions. And it offers the generation of time-limited secure links.

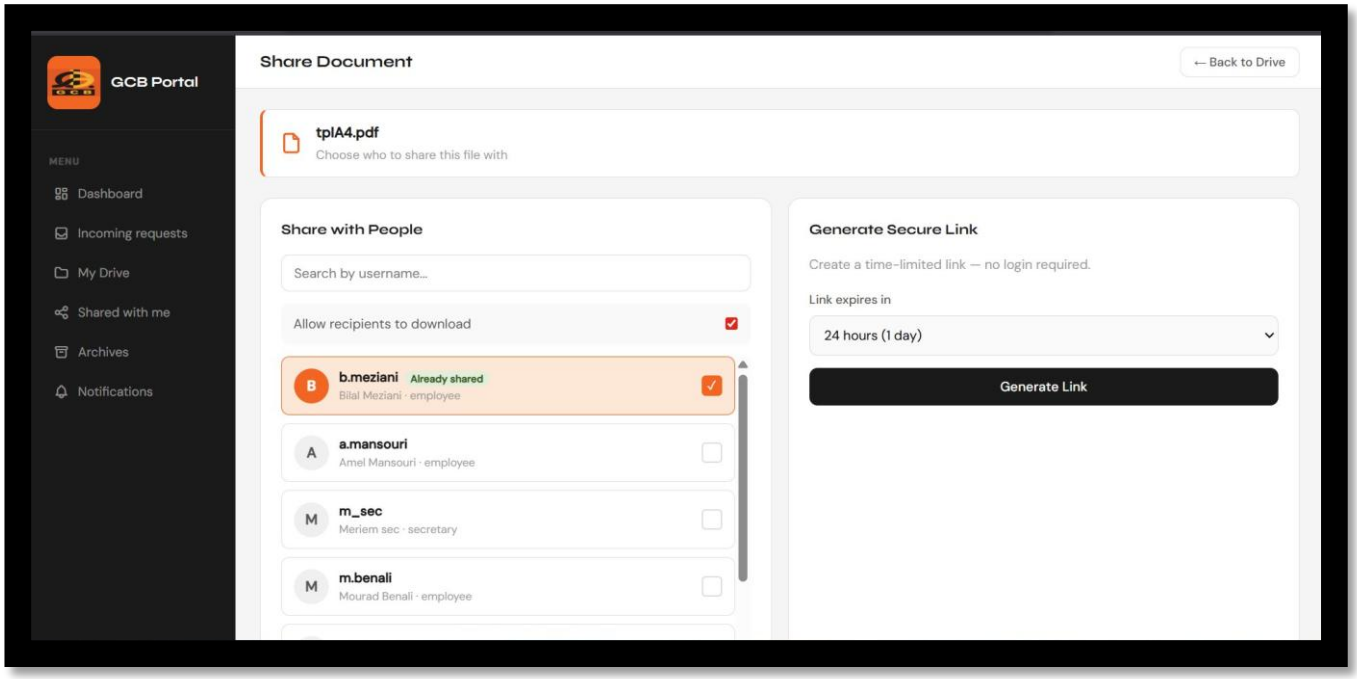


Figure4.23: Share with people

●Notifications:

Displays notifications specific to the employee's activity, primarily centered around documents shared with them by other users.

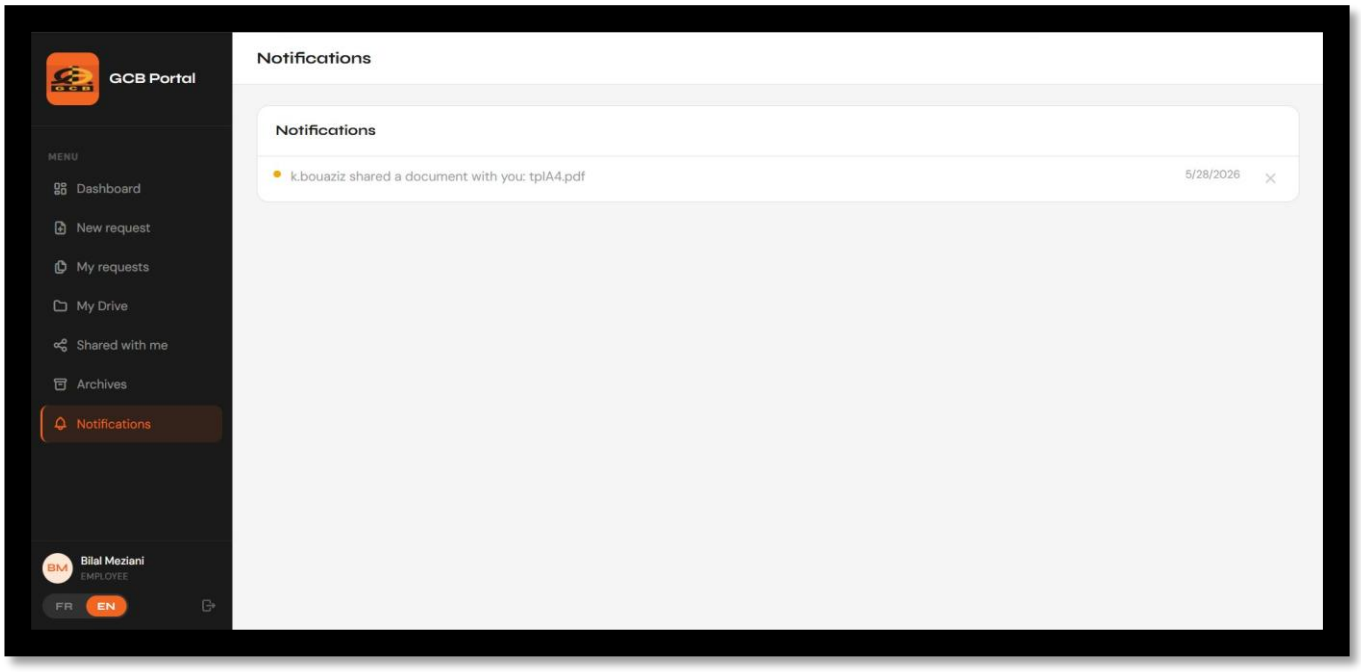


Figure4.24: Employee Notifications

4.3. Secretary Space:

●Secretary Dashboard:

The secretary dashboard presents a functionally differentiated view. Summary indicators display the total volume of received requests, pending Requests and current drive storage consumption. (My Drive, shared with me and Archives are the same for the employee).

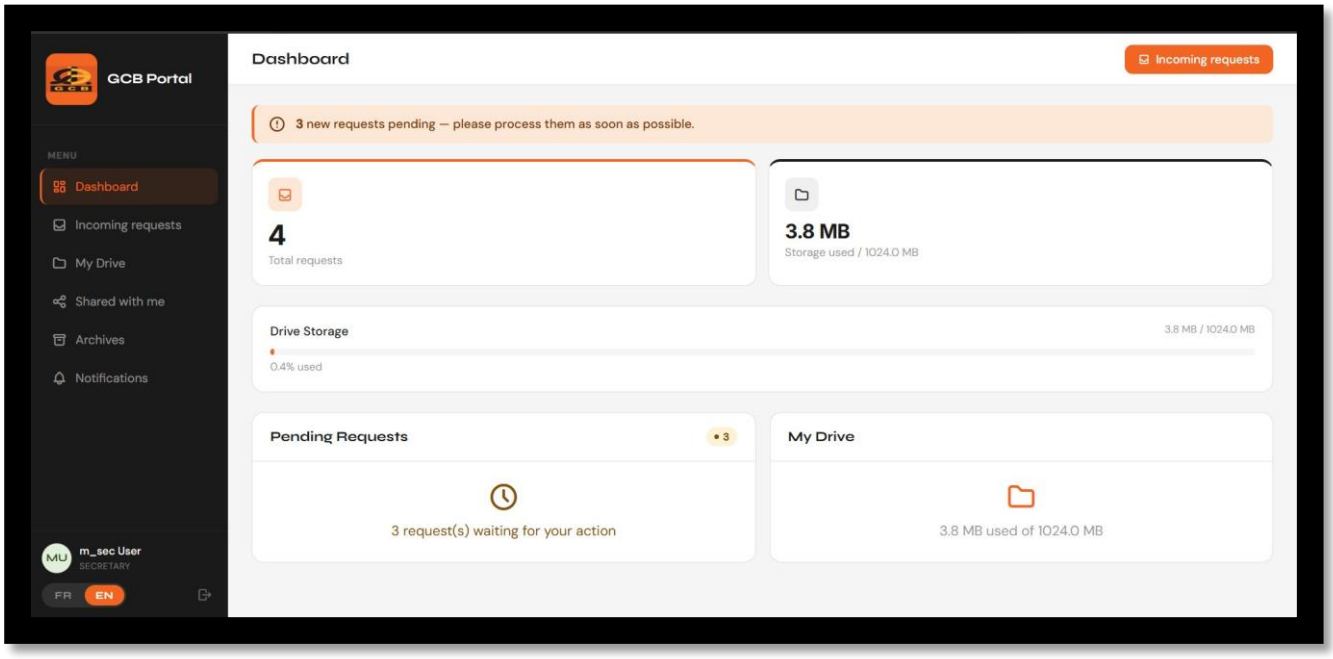


Figure4.25: Secretary Dashboard

●Incoming Request:

The Incoming Requests interface constitutes the primary operational workspace for secretarial users, centralising all employee-submitted requests awaiting administrative processing. A summary of total, pending and received requests. The all requests present each submission with its associated employee identifier, request type, submission date, and current status. For each pending entry, the secretary may perform two actions: Mark as Received, and Print.

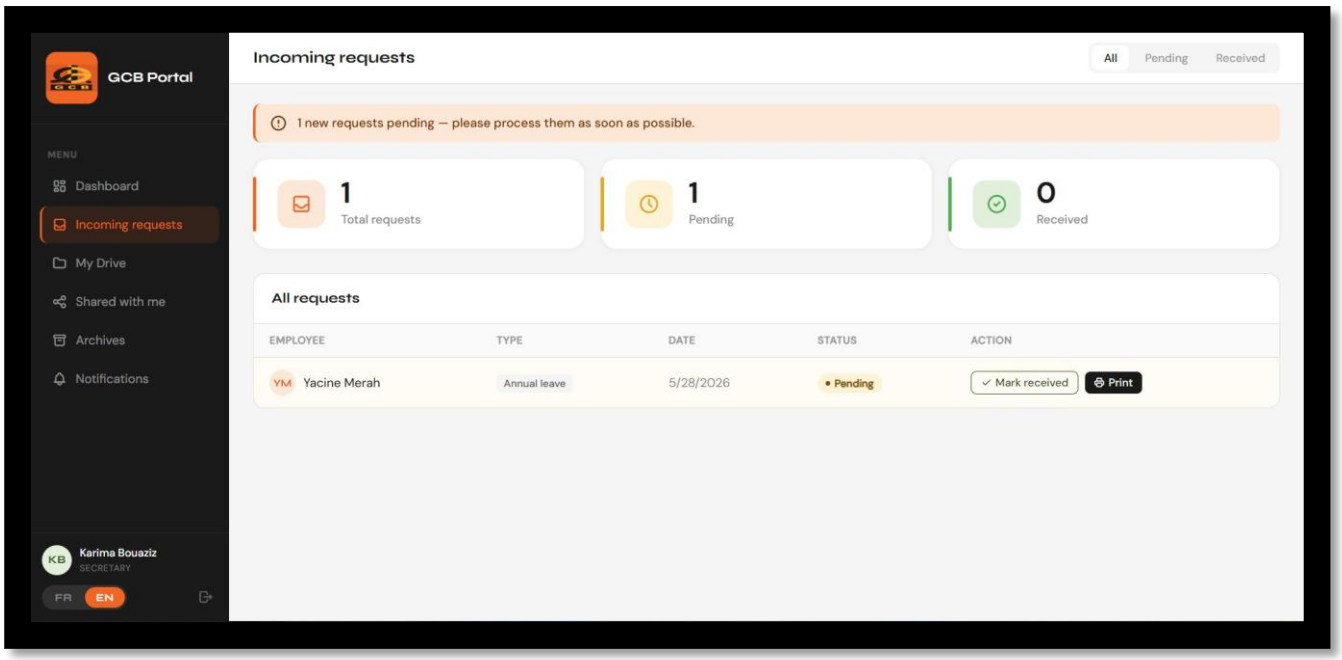


Figure4.26: Incoming Request

●Notification:

Displays real-time alerts aggregating all operations relevant to the secretary's role, including incoming requests and document sharing events.

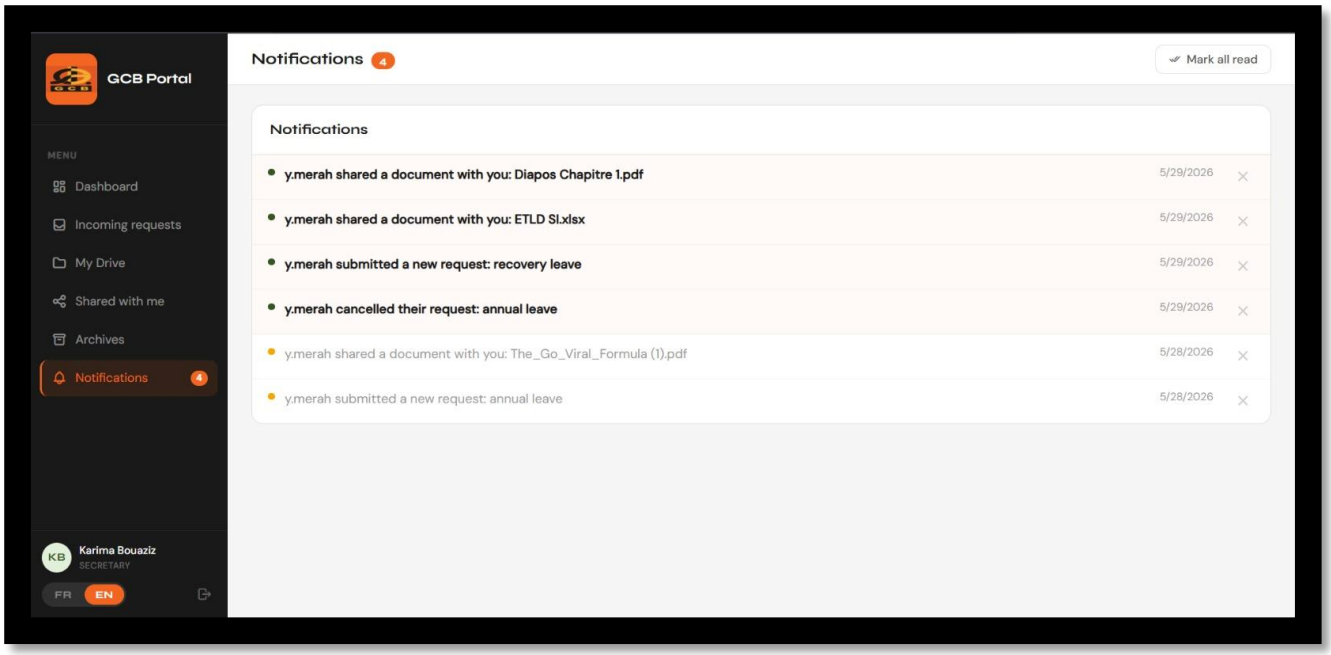


Figure4.27: Secretary Notifications

4.4. Admin Space:

●Admin dashboard:

The administration dashboard serves as the central control interface of the GCB Portal, providing administrators with an immediate summary of the system's current state. While two complementary panels presenting recently registered users alongside their roles and activation status, and a real-time feed of the most recent system log entries.

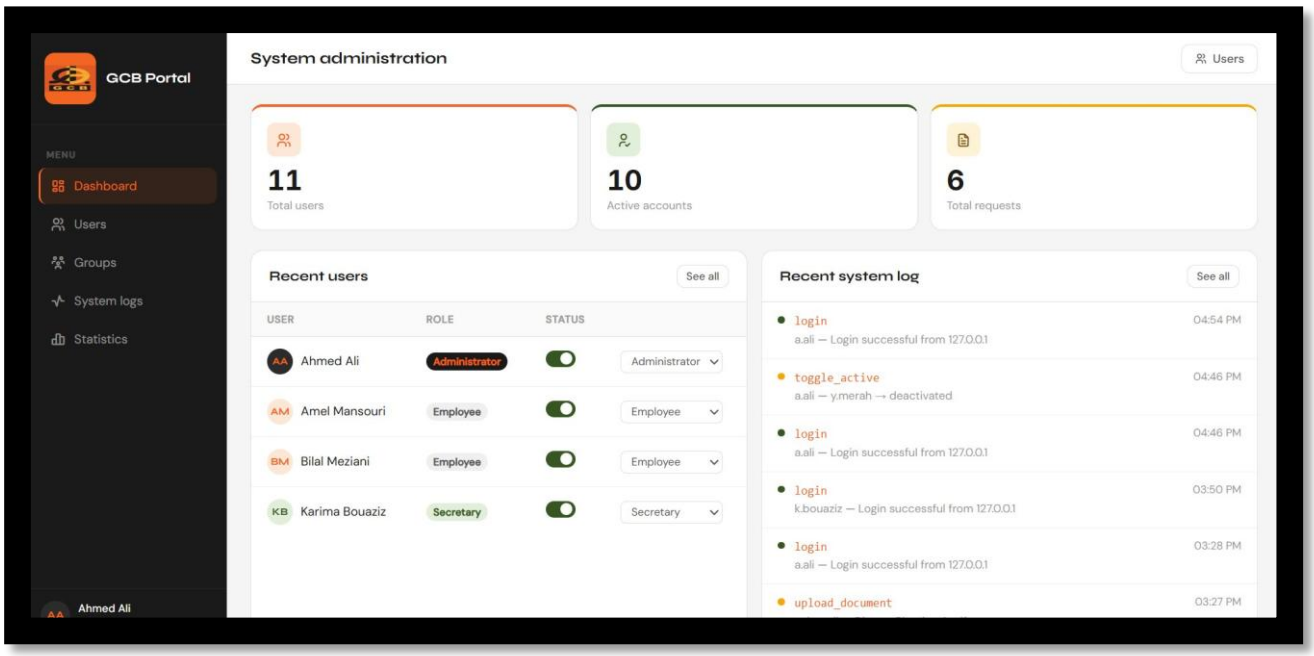


Figure4.28: Admin Dashboard

●User management:

The user management interface presents a structured directory of all registered accounts, detailing each user's role, group affiliation, activation status, and allocated storage quota.

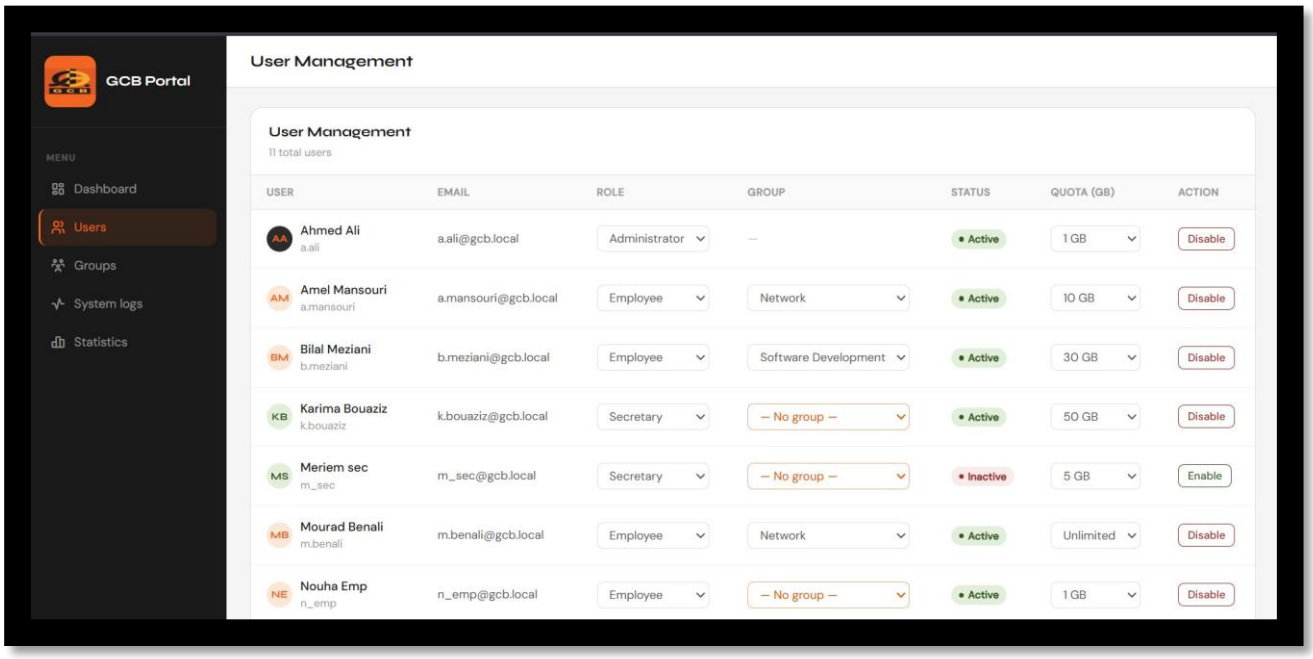


Figure4.29: User management

●Group management:

The group management interface allows administrators to organize users into functional units, each assigned a designated secretary. Groups are displayed with their respective members, and administrators may create, modify, or delete groups through the provided controls.

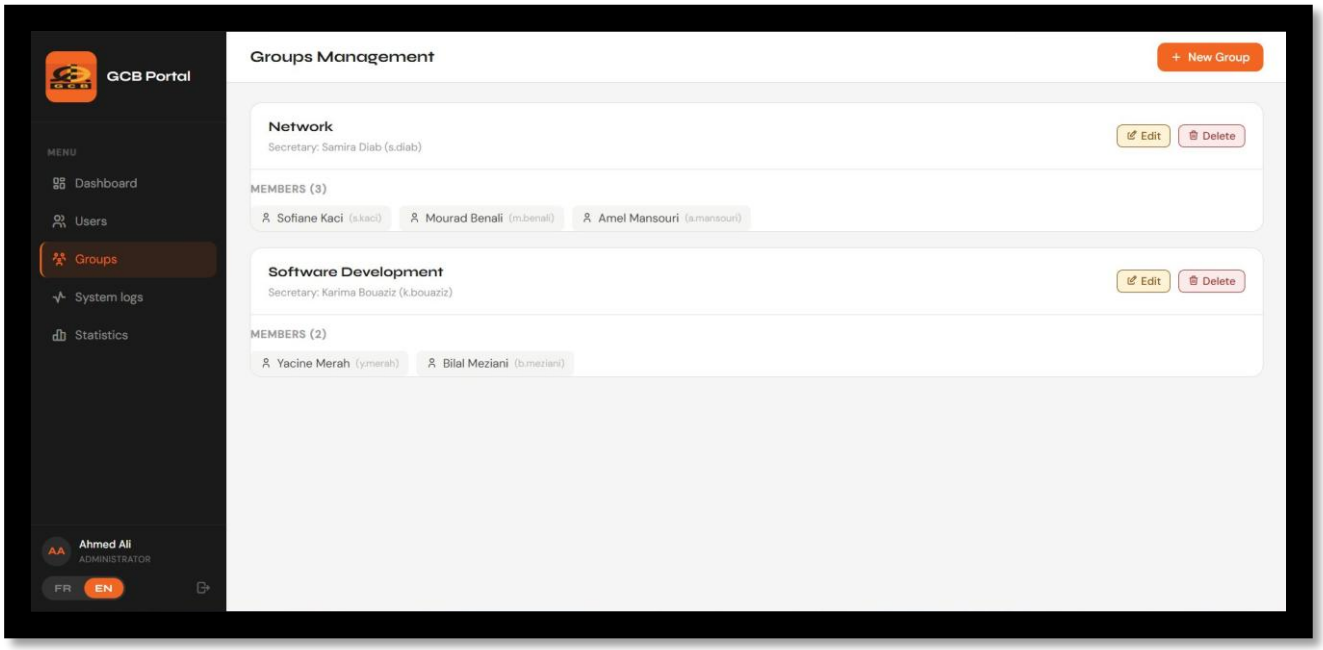


Figure4.30: Group management

●System Logs:

The system logs interface provides a chronological audit trail of all user activity within the platform. Each entry records the timestamp, username, action type (login, submit_request, archive_document, login_failed...), and associated details, enabling administrators to monitor system usage.

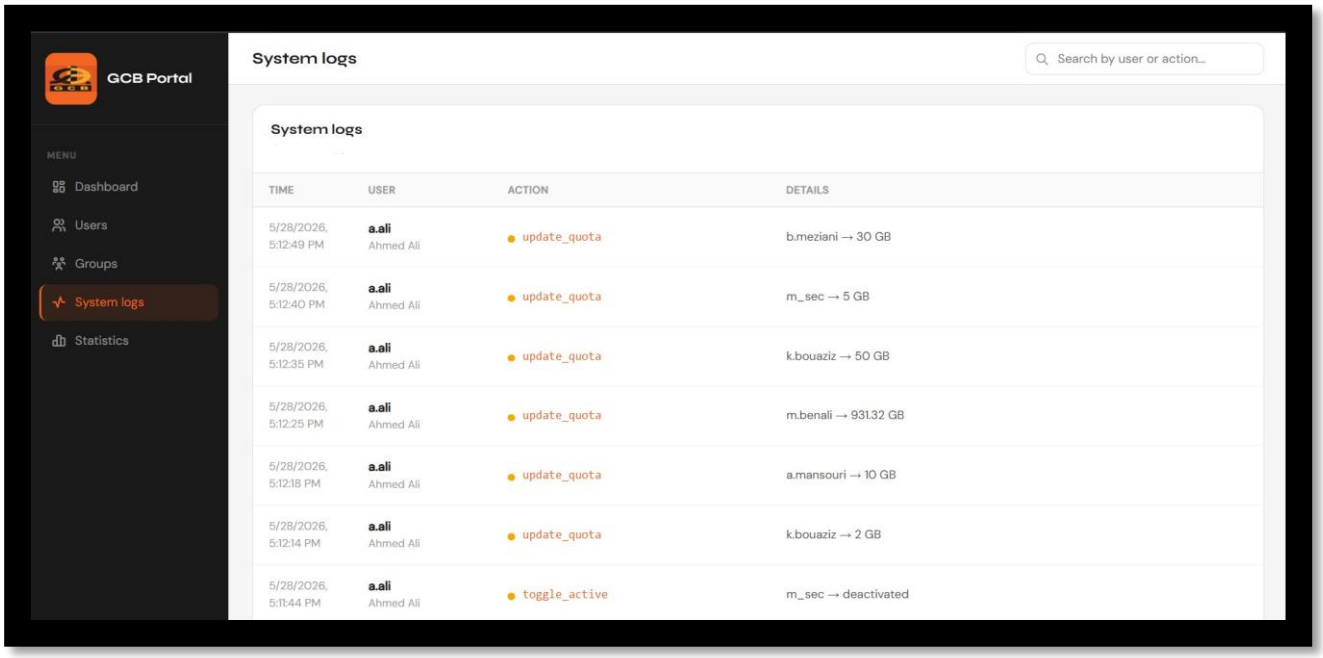


Figure4.31: System log

●Statistics:

The statistics interface offers a consolidated overview of key performance indicators, including total user accounts, active and inactive account ratios, and request processing status.

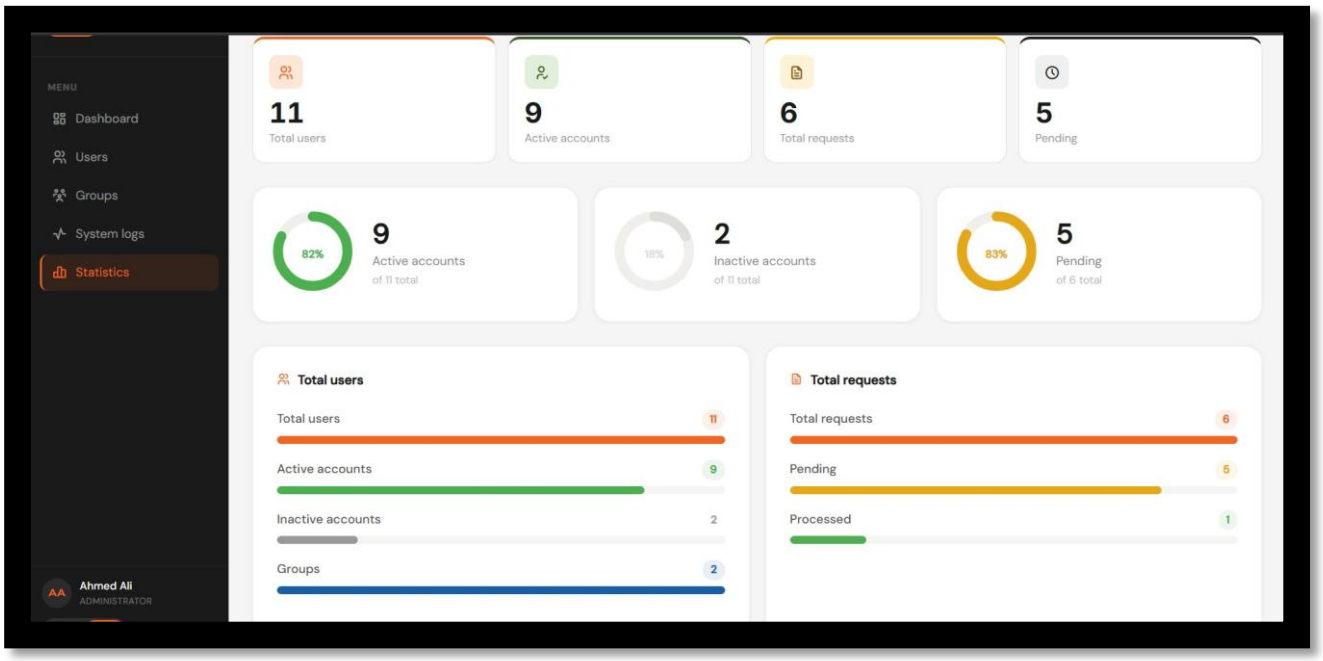


Figure4.32: Statistics

conclusion:

This chapter presented the complete implementation of the GCB administrative management system. The application was developed using React and Laravel, deployed in a virtualized windows server 2022 environment, and integrated with the company's Active Directory for secure authentication. The system covers all defined functional requirements, from administrative request submission to secure document sharing and archiving.

General conclusion

This graduation project successfully addressed a significant operational challenge within GCB Spa by designing and implementing a secure, scalable, and efficient three-tier web application aimed at replacing inefficient paper-based administrative procedures. Conducted within the IT Department at GCB's headquarters in Boumerdes as part of a Licence Academic, the proposed solution contributes to the modernization of request management and document handling processes for an organization comprising approximately 6,000 employees.

Built upon a robust technological stack integrating React.js (Vite), Laravel 11, and MySQL, the developed system enables the complete digitization of administrative workflows through dedicated interfaces tailored to Employees, Secretaries, and Administrators. Beyond fulfilling the primary functional requirements, the project adopts enterprise-level practices to ensure reliability, security, and maintainability. These include seamless integration with Active Directory through the LDAP protocol for centralized authentication and automated user synchronization, as well as the implementation of a robust Role-Based Access Control (RBAC) mechanism to enforce authorization policies and prevent privilege escalation. Furthermore, in order to simulate a realistic enterprise infrastructure, virtualization technologies based on Windows Server 2022 and Hyper-V were employed to deploy and configure Active Directory services alongside a dedicated server environment for system validation and testing.

Despite the successful achievement of the project's initial objectives, several perspectives for future enhancement were identified. The system already supports configurable per-user storage quotas managed by the administrator, as demonstrated in the user management interface. Likewise, integrating encryption mechanisms for sensitive files stored on the server filesystem would reinforce data confidentiality and complement the HTTPS protocols already ensuring secure data transmission. In addition, extending the notification system to support email-based alerts would improve communication efficiency by enabling users to receive real-time updates regarding request progress, even when disconnected from the platform.

In conclusion, the modular architecture adopted within the Laravel backend guarantees the extensibility and long-term maintainability of the platform, facilitating the seamless integration of new request categories and additional hierarchical validation procedures in response to evolving organizational needs. Therefore, this project not only provides a complete and deployable solution to an existing administrative problem but also represents a meaningful contribution to GCB's ongoing digital transformation strategy, establishing a reliable technological foundation for the future evolution of the company's internal information systems.

Bibliography:

https://gcb.dz/
https://gcb.dz/PresentationGCB
https://gcb.dz/la-societe/presentation/
https://gcb.dz/Organisation
(Those presents, GCB Presentation) [visited on, March the 12th]

https://www.omg.org/uml/
(This presents the 3-tier System architecture) [visited on, May 22nd]

https://www.ibm.com/think/topics/three-tier-architecture
(This presents the 3-tier System architecture) [visited on, May 22nd]

https://dbdiagram.io/home
(This presents the justification of dbdiagram use) [visited on, May 22nd]

https://docs.npmjs.com/about-npm
(This presents the justification of npm use) [visited on, May 22nd]

https://nodejs.org/learn/getting-started/introduction-to-nodejs
(This presents the justification of Node.js use) [visited on, May 22nd]

https://getcomposer.org/doc/00-intro.md
(This presents the justification of Composer use) [visited on, May 22nd]

https://laragon.org/why-laragon
(This presents the justification of Laragon use) [visited on, May 26nd]

https://docs.usebruno.com/introduction/getting-started
(This presents the justification of Bruno use) [visited on, May 22nd]

https://www.drawio.com/search?search=why+draw.io
(This presents the justification of draw.io use) [visited on, May 22nd]

https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/
(This presents the justification of Hyper-V use) [visited on, May 22nd]

https://code.visualstudio.com/docs/editor/whyvscode
(This presents the justification of VS code use) [visited on, May 22nd]

https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2022
(This presents the justification of windows server 2022 use) [visited on, May 22nd]

https://www.powerdesigner.biz/EN/
(This presents the justification of PowerDesigner use) [visited on, May 22nd]

https://www.lucidchart.com/pages
(This presents the justification of Lucidchart use) [visited on, May 22nd]

https://httpd.apache.org/
(This presents the justification of Apache use) [visited on, May 20nd]

https://axios.rest/pages/advanced/interceptors
(This presents the justification of Axios use) [visited on, May 20nd]

https://learn.microsoft.com/en-us/entra/architecture/auth-ldap
(This presents the justification of LDAP use) [visited on, May 20nd]

https://www.mysql.com/why-mysql/
(This presents the justification of MySQL use) [visited on, May 20nd]

https://www.php.net/intro
(This presents the justification of PHP use) [visited on, May 20nd]

https://laravel.com/docs/13.x/csrf
https://laravel.com/docs/13.x/eloquent
https://laravel.com/docs/13.x/authentication
(This presents the justification of Laravel use) [visited on, May 20nd]

https://vite.dev/guide/why.html
(This presents the justification of Vite use) [visited on, May 20nd]

https://react.dev/learn/describing-the-ui
https://react.dev/learn/adding-interactivity
(This presents the justification of React use) [visited on, May 20nd]