People's Democratic Republic of AlgeriaMinistère de Higher Education and Scientific Research University M'hamed Bougara - Boumerdes

Faculte of Sciences
Department of Computer Science

**Field** : Mathematics and Computer Science
**Program** : Computer Science
**Specialty** : IT systems
*N° de l'Arrêté d'habilitation de la spécialité : arrêté n °872 du 26/07/2016*

***Mémoire de fin d'études en vu de l'obtention du Diplôme de Licence Académique***

# *Thème*

# Conception et réalisation d'une application de gestion des PV rectificatifs

***Presented by:***

***BEROUAKEN Chaima***

***OUCHERIF Rania***

***SAHI Fatma Zahra***

***TOBAL Rekia***

***Internship completed at : M'hamed Bougara University - Boumerdès***

***Supported on …../…./ ….. Before the jury composed of***
Drifa Hadjidj : Examiner
Lounas Razika: Supervising

# Thanks

*I would like to express my sincere gratitude to all the people who contributed, directly or indirectly, to the completion of this dissertation.*

*First of all, I would like to warmly thank my supervisor for their guidance, valuable advice, availability, and continuous support throughout this work.*

*I would also like to thank the members of the jury for the honor they have given us by evaluating this dissertation.*

*My thanks also go to all our teachers for the knowledge and education they provided during our academic journey.*

*Finally, I would like to express my deepest appreciation to my family, friends, and everyone who supported and encouraged me throughout this period.*

## *Dedication*

*I dedicate this work to my beloved family, whose endless love, support, and encouragement have always been my greatest source of strength.*

*To my parents, for their sacrifices, patience, and unconditional support throughout my academic journey.*

*To my friends and all those who stood by me and motivated me during the completion of this work.*
*And to everyone who believed in me and encouraged me to achieve my goals.*

*Sahi fatma zahra*

## *Dedication*

*This work is dedicated to my dear parents, whose love, prayers, and sacrifices have always guided me toward success.*

*To my family, for their constant encouragement and support during every step of my studies.*

*To my friends, who shared with me both the difficult and joyful moments throughout this journey.*

*And to all those who inspired me to keep moving forward and never give up on my dreams.*

*BEROUAKEN Chaima*

## *Dedication*

*I dedicate this humble work to my family, especially my parents, for their endless love, patience, and support.*

*To my brothers and sisters, for their encouragement and motivation throughout my studies.*

*To my friends, for their kindness, support, and unforgettable moments.*
*And to everyone who helped me, believed in me, and encouraged me to achieve this success.*

*OUCHERIF Rania*

## *Dedication*

*With great pleasure, I dedicate this work to:*

*My beloved parents, for their unconditional love, endless support, and sacrifices throughout my life.*

*My family, for their encouragement, patience, and constant motivation.*

*My friends, for their support, understanding, and the wonderful moments we shared.*

*And to everyone who contributed, directly or indirectly, to the completion of this work.*

*TOBAL Rekia*

# Abstract

*The increasing complexity of administrative processes in Algerian universities has highlighted the need for digital solutions. This project presents the design and development of an Android mobile application for managing grade correction reports (PV Rectificatifs) at the University of Boumerdes. The application enables students to submit rectification requests, teachers to review and validate them, and administrators to generate official PV documents in PDF format. The system was modeled using UML and implemented using Kotlin, Python/Flask, and MySQL.*


***Keywords:*** *Android, Flask, Kotlin, MySQL, PV Rectificatif, UML*

Academic year: 2025/2026

Chapitre 1:Digital Collaboration and Collaborative Platforms……..

Introduction .....1

What Is DigitalCollaboration ..... 1

Elements of Collaborative Work ..... 1.

Comparison of Work Approaches ..... 2:

Characteristics of Collaborative Work.....3.

KeySuccess Factorsfor Effective Collaboration..... 3.

Collaborative Platforms Examples .....4

Conclusion.....4

Chapitre2:Digital Collaboration in theAlgerian University Context

Introduction .....5

*DigitalTransformation in Algerian Universities*.....*5*

Current Situation .....5.

Actors andWorkflow ofPVRectificatifs .....6 Problems

Encountered...., .....6

Proposed Solutions, ..... 7

Conclusion.....,.....8.

Chapter3:SystemSpecificationand Requirements Analysis

Introduction .....9

Definition of UML .....

9 Requirements Analysis ..... 9

Functional Requirements..... ..... ,

.....9.

Non-Functional Requirements.....,..... 10

Functional Analysis..... 10

Use Case Diagrams ..... 10

Identification of Actors..... 10

GeneralUseCase Diagrams ..... 11
Administrator_UseCase_Diagrams ..... 12
Teacher_UseCase_Diagrams ..... 12
Student Use Case Diagram ..... 12
Text Description of Use Cases Use Cases ..... 13
Authentication ..... 15
Student Add Request ..... 13
View Request ..... 13
Delete Request ..... 14
Add Rectification Document ..... 14
Validate Rectification ..... ,15
Sequence Diagram ..... 15
Authentication SequenceDiagram ..... 15
Add Request (Student) ..... 16
Add Rectification Document (Teacher) ..... 17
Validate Rectification (Admin) ..... 18
Conclusion ..... 18
Chapter 4: System Design. ..... 19
Introduction ..... 19
Class Diagrams. ..... 19
Data Dictionary ..... 20
The Proposed Class Diagram ..... 21
Passage to the Relational Model ..... 22
Relational Model ..... 19
Detailed Sequence Diagrams ..... 24
Conclusion ..... 30
**Chapter 5: Implementation.**
Introduction ..... 37
Development Environment ..... 37
Hardware Environment ..... 37
Software Environment ..... 37
Technologies and Programming Languages ..... 37
Kotlin (Android) ..... 37
Python / Flask (Backend) ..... 38
MySQL (Database) ..... 38
REST API Architecture ..... 38
Application Interfaces ..... 38

Authentication Interface Login Screen ..... 38
Student Dashboard Figure ..... 39.
Grade Rectification Request Interface..... 39
Send to Teacher Interfa..... 40..
Grades and Status List Figure ..... 41
Request History Interface ..... 41
Official PV Document PDF Generation Conclusion..... 42.

# *General Introduction*

The rapid evolution of information and communication technologies has significantly transformed the way organizations and institutions manage their activities and collaborate. In recent years, digital collaboration has become an essential component in improving communication, coordination, and productivity within professional and academic environments. Universities, in particular, increasingly rely on digital systems to facilitate administrative and educational processes, ensure efficient information management, and support collaboration among students, teachers, and administrative staff.

Despite the progress achieved in the digitalization of higher education, several administrative procedures in Algerian universities are still handled through traditional methods based on paper documents, emails, and non-integrated digital files. Among these procedures is the management of corrective reports (PV rectificatifs), which are used to correct errors related to students' grades or academic records after the publication of examination results. The current management of these reports often suffers from multiple limitations such as slow processing, data entry errors, lack of traceability, document loss, and communication difficulties between stakeholders.

To overcome these challenges, there is a growing need for collaborative digital platforms capable of centralizing information, automating workflows, and improving coordination among all actors involved in the correction process. Such systems can enhance transparency, reduce administrative workload, facilitate remote access, and ensure efficient tracking of requests from submission to validation and archiving.

The main objective of this project is therefore to design and implement a collaborative digital application dedicated to the management of PV rectificatifs within university institutions. The proposed solution aims to simplify the submission and processing of rectification requests while improving communication and collaboration between students, teachers, and administrators. The application also seeks to ensure better traceability, secure data management, and automation of administrative tasks.

This dissertation is organized into five chapters:

- The first chapter presents the theoretical foundations of digital collaboration and collaborative platforms, including their characteristics, processes, and importance in modern information systems.
- The second chapter analyzes the current management of PV rectificatifs in Algerian universities, identifies the main challenges of the existing system, and introduces the proposed collaborative solution.
- The third chapter focuses on system specification and requirements analysis using UML diagrams and functional analysis techniques.
- The fourth chapter presents the system design, including the class diagram, relational model, and sequence diagrams.
- Finally, the fifth chapter describes the implementation phase of the application, the technologies used, and the main user interfaces developed for the system.

Through this work, we aim to contribute to the modernization of administrative management in Algerian universities by proposing a reliable and efficient collaborative platform adapted to the needs of academic institutions.

# Chapitre1 : Digital Collaboration and Collaborative Platforms

## Introduction:

This chapter presents the theoretical foundations of digital collaboration and collaborative platforms, which are essential concepts in modern information systems. It first introduces the notion of digital collaboration and explains the main elements that support collaborative work, including communication, coordination, and collaborative production. The chapter also describes the main characteristics of collaborative work and highlights the differences between collaborative and individual work. Finally, it introduces the concept of collaborative platforms and their role in facilitating interaction and team
within organizations. This chapter therefore provides the theoretical background necessary for understanding collaborative digital systems.

# 1. Digital Collaboration:

Digital collaboration is the use of digital tools, apps, and systems that let people work together from anywhere, in real-time or asynchronously. It replaces traditional face-to-face methods with digital channels, such as shared documents, messaging platforms, and videocalls, to keep teams aligned and productive regardless of where they are [1]. Collaborative work requires a set of processes that allow multiple people to work together efficiently. These processes can be grouped into three essential elements: communication, coordination, and production.

## . Elements of Collaborative Work

### a. Digital Communication :

Digital Communication is the exchange of information using digital tools like email, messaging, video calls, and collaborative platforms [1] (e.g., Microsoft Teams, Slack, Zoom, Gmail). It enables team members to share updates, clarify tasks, and provide feedback in real time or asynchronously. Effective digital communication support scoordination and productivity in collaborative work.

### b. Digital Coordination :

Digital Coordination is the process of organizing tasks, schedules, and responsibilities among team members using digital tools and platforms [1] (e.g., Trello, Asana, Monday.com, Jira). It ensures that work is sequenced efficiently, deadlines are met, and dependencies are managed. Effective digital coordination reduces confusion and improves overall team performance..

### c. Collaborative Production:

Collaborative Production is the process where team members jointly create, edit, and finalize work using digital tools. It relies on shared responsibility, coordination, and ongoing decision-making to produce consistent and high-quality outputs [2] (e.g., Google Docs, Notion, Confluence, SharePoint). Collaborative production supports both real-time and asynchronous teamwork.

# 2.Comparison of Work Approaches :

To better understand the different approaches to organizing work, the following table compares individual, cooperative, and collaborative work based on several criteria such as interaction, task organization, and dependency between participants.

| Criteria | Individual Work | Cooperative Work | Collaborative Work |
|---|---|---|---|
| Responsibility | Each person is accountable for their own work | Each member responsible for their sub-task; overall outcome shared | Team is collectively responsible for the final result |
| Communication | Minimal or none | Limited, mainly to coordinate hand offs | Continuous exchange of ideas and feedback |
| Coordination | Usually absent | Required for dividing tasks and integrating results | Essential for scheduling, organizing tasks, and aligning contributions |

| | | | |
|---|---|---|---|
| Production | Output belongs solely to the individual | Sub-tasks are completed independently and combined | Joint creation; all contributions integratedinto shared output |
| Flexibility / Scheduling | Works on own schedule | Moderate: must meet deadlines for sub-tasks | Requires jointplanning and real-time coordination |

## Analysis:

Individual, cooperative, and collaborative work differ mainly in interaction and responsibility.

- **Individual work**: each person works independently and is fully responsible for their own results. [1]
  *Example:* astudent researching and writing a literature view alone.
- **Cooperative work**: tasks are divided among team members who work independently and later combine their results.
  *Example:* a team conducting different experiments separately in a lab, then merging the results in a single report.
- **Collaborative work**: all participants actively work together,shareres ponsibility, and co-create the outcome.
  *Example:* a group of students designing asurvey together, discussing questions, analyzing responses collectively, and deciding on conclusions as a team. [1]

These distinctions help in choosing the right tools and processes for teamwork. Digital platforms can support collaboration by facilitating communication,coordination, andshared production.

## 2. Characteristics of Collaborative Work:

Collaborative work has several key features that distinguish it from individual work:

- **Participation of multiple individuals:** Collaborative work relies on a group of people contributing together to achieve a shared goal, whereas individual work depends on a single person handling all responsibilities(example: Students jointly collecting research data)
- **Exchange of ideas and knowledge:** Collaborative work allows team members to share information and expertise, which helps develop ideas and improve thequality of results.
- **Shared responsibilities:** Tasks are distributed among team members so that each person contributes to the overall completion of the work.
- **Continuous communication:** Collaborative work requires ongoing interaction between participants through information sharing and coordination.
- **Coordination and organization:** Effective collaboration requires organizing efforts and coordinating tasks to ensure alignment and achievement of shared objectives.
- **Enhancement of creativity and problem-solving:** Collaborative work leverages the diversity of skills and experiences within the team, helping to generate more effective solutions compared to individual work.

## 3. Collaborative Platforms:

## 4. Exemples :

Collaborative platforms are digital systems designed to support teamwork by providing shared environments where users can interact, coordinate tasks, and produce collective outputs. These platforms integrate essential functionalities such as communication tools, project management systems, document sharing, and access control mechanisms [3]. Collaborative platforms improve transparency, efficiency, and team coordination, enabling participants to work together regardless of geographical location.

### 4.1 *Exemples of Collaborative Platforms:*

- **Microsoft Teams:** Provides chat, meetings, and filesharing; widely used in universities and companies to facilitate collaboration and group projects.
- **Slack: Offers** specialized communication channels, project management integration, and remote work support.
- **Google Workspace (Docs, Sheets, Drive):** Allows real-time co-editing of documents, spreadsheets, and presentations, supporting both synchronous and asynchronous collaboration.
- **Trello / Asana:** Task management platforms for assigning responsibilities, tracking progress, and ensuring effective coordination among team members

## 5. Key Success Factors for Effective Collaboration:

Effective collaborative work depends on several key success factors that ensure teams achieve their goals efficiently and harmoniously. These include:

- **Clear Roles and Responsibilities:** Each team member should understand their tasks and how they contribute to the overall outcome.
- **Continuous Communication: Open** and timely sharing of information, feedback, and ideas aligns team efforts.
  *Example:* Remote software development teams use Slack or Microsoft Teams for daily stand-up updates and code reviews.
- **Trust and Psychological Safety:** Team members must feel safe to speak, suggest ideas, or question decisions without fear.
  *Example:* Google's Project Aristotle found that high-performing teams share psychological safety, enabling better collaboration. [4].
- **Use of Appropriate Tools:** Digital platforms should facilitate coordination, task management, and joint production.
  *Example:* Trello or Asana for task tracking, Google Docs for real-time document editing.
- **Structured Processes:** Well-defined workflows help organize tasks, reduce conflicts, and integrate contributions.

These factors provide a practical framework for designing and implementing collaborative systems in both academic and professional contexts

## Conclusion

This chapter presented the fundamentals of collaborative work and digital platforms, high lighting key processes, characteristics, and practical examples. These concepts provide the foundation for the next chapters, which will detail the design and implementation of digital systems to support effective teamwork.

# *Chapitre2 : Digital Collaboration in the Algerian University Context*

# Introduction:

In the previous chapter, we presented the theoretical framework of collaborative work and its importance in educational management. However, understanding theory alone is not sufficient without examining its practical application. Therefore, this chapter analyzes the current management of corrective reports (PV rectificatifs) in university institutions. It highlights the main challenges associated with traditional administrative procedures and introduces a solution designed to improve accuracy, traceability, and collaboration among stakeholders.

## 1. Digital Transformation in Algerian Universities:

Recent research on digital transformation in Algerian universities highlights the critical role of digital technologies in enhancing both academic and administrative management. Studies involving students and faculty members show that digital platforms and information systems substantially improve the organization of educational processes, accelerate data handling, and strengthen communication among university stakeholders. The COVID-19 pandemic has further emphasized the need for digital solutions, as universities were forced to adopt remote learning and online administrative procedures to maintain educational continuity. Despite these advancements, many administrative processes still rely on traditional methods, resulting in inefficiencies and delays. Consequently, scholars advocate for the development of specialized information systems to modernize university administration, improve operational efficiency, and ensure greater transparency. [5][6][7]

## 2. Current Situation :

In many educational institutions, the management of corrective reports (PV rectificatifs) is carried out through administrative procedures that rely mainly on paper documents and basic digital tools. These corrective reports are generally prepared when an error is identified in students' grades or academic information after the publication of examination results.

Several stakeholders are involved in this process, including teachers, the academic administration, and the department responsible for academic affairs. When an error is detected, the concerned teacher prepares a corrective report that includes the relevant information about the student, the course or module, the initial grade, and the corrected grade.

Once the corrective report is prepared, it is submitted to the academic administration where it is recorded and verified to ensure the accuracy of the provided information. After the verification step, the document is forwarded to the responsible authority for validation and official approval.

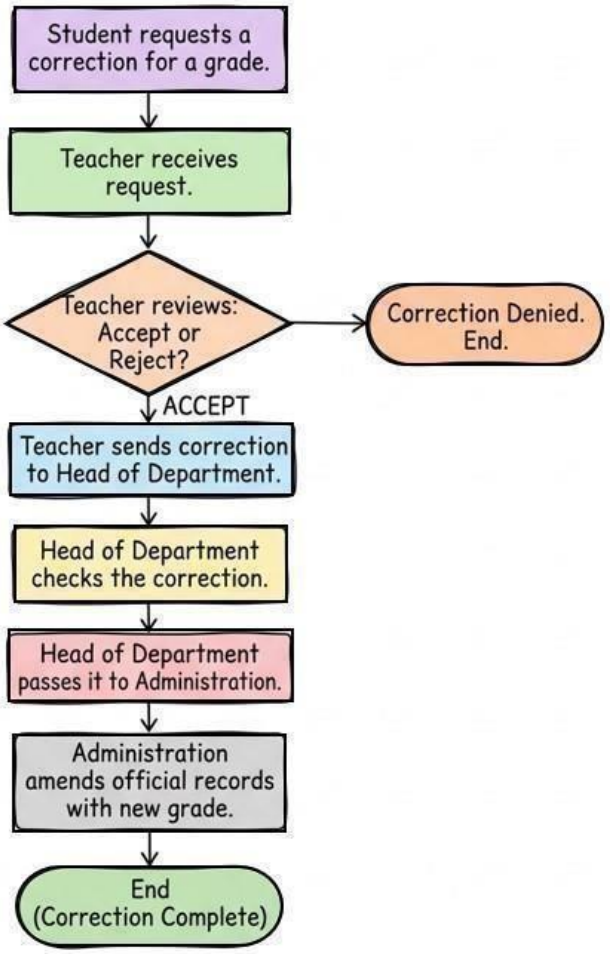
Finally, the approved corrective report is archived within the institution's administrative records for documentation and futurereference. In many cases, these reports are stored

either as paper documents or as digital files such as PDF documents or Excel spreadsheets, organized according to academic year, department, or course.

## 3. Actors and Workflow of PV Rectificatifs

- The Student initiates the request.
- The Teacher checks the requestand adds correction suggestions.
- The Head of Department reviews and approves the corrections.
- The Administrative Staff finalizes the PV, archives it, and informs the Student

## Workflow of PV Rectificatifs



## 4.Problems Encountered:

Despite the existence of administrative procedures for managing corrective reports (PV rectificatifs) in educational institutions, the current system still relies largely on manual processing and non-integrated documents. This management approach can lead to several difficulties that affect the efficiency and reliability of handling these reports.
Among the main problems observed in the current system are the following:

- Loss of paper documents or difficulty locating them after a certain period of time.
- Difficulty searching for previously archived reports due to the absence of a centralized system for organizing them.
- Frequent data entry errors, such as mistakes in student names, grades, or course modules, resulting from manual data input.
- Slow processing of requests because the corrective report must pass through several administrative steps before validation.
- Lack of clear traceability of modifications made to the reports.
- Large number of emails and communication overall which makes tracking requests and their status more complicated.

These challenges highlight the need for an oreorganized system capable of improving the management, processing, and archiving of corrective reports.

## 4. Solutions Proposed

To address the problems identified in the management of PV rectificatifs, it is proposed to develop a collaborative digital platform dedicated to handling correction requests. This platform will centralize all activities in asingle system, improving transparency, traceability, and efficiency.

The platform will allow students to submit correction requests when they detect errors in their marks. Teachers can verify and validate these requests, while administrative staff will update the official PV records through the system. Role-based access ensures that each user performs actions according to their responsibilities, enhancing security and accountability (“Turban, Pollard & Wood, 2018”).

Key features of the proposed system include:

- **Digitization of the entire process:** Eliminating paper-based procedures and manual communication.
- **Centralized data base:** All requests are stored in a single location for easier management and retrieval.
- **Real-time tracking:** Users can follow the status of their requests from submission to approval.
- **Audit trail:** Recording all modifications with user identity, date, and time to ensure transparency.
- **Work flow automation:** Automating stages from request registration to archiving to reduce delays and errors.
- **Automated notifications:** Sending alerts to stakeholders when anew requestis submitted or a stage is completed.
- **Standardized digital formats:** Replacing multiple media (paper, Excel,PDF) to ensure consistency.
- **Enhanced collaboration:** Allowing all departments to work on asingle platform, reducing conflicts and improving coordination.
- **Secure remote access:** Enabling authorized users to access the system from anywhere.
- **User-friendly interface: Simplifying** the submission, tracking,and management of requests.
- **Efficient search and archiving:** Facilitating quick retrieval of past reports and secure storage.
- **Early warnings:** The system can detect delays in processing requests or recurring issues and send an alert to university administration.
- **Explain the reasons** (acceptance/rejection) **:** allow the instructor to add comments, enabling the student to understand the decision and improve future applications.
- **Follow the entire event remotely:** without having to be physically present.
- **One-click approval or rejection:** Relying on manual processes increases the administrative burden on professors, especially given the high volume of requests. Therefore, a simple interface should be provided that displays requests in an organized manner and allows for one-click approval or rejection, there by saving time and effort.
- **Present all necessary information in an organized manner within the request:**
- Requests may contain incomplete or unclear information, which complicates the verification process and decision-making. Therefore, all necessary information such as: (student's name, original grade, proposed grade, reason for the change) must be

- included

  to ensure an accurate decision is made ..

By integrating these solutions, the proposed platform addresses the key limitations of the traditional system, reduces administrative workload, minimizes errors, and ensures a transparent, efficient, and reliable PV rectification process.

# Conclusion

In this chapter, we examined the current practices of managing PV rectificatifs in university institutions, identifying the main actors involved in this process and analyzing how corrective reports are currently handled. We also highlighted the main challenges related to traditional management methods, such as slow processing and limited traceability of modifications. Based on this analysis, a digital collaborative solution was proposed to improve the management of these processes and facilitate communication between the different stakeholders. The following chapters will present the system requirements, design, and implementation of the proposed application.

# Chapter3:System Specification an

# Requirements Analysis

## III.1 Introduction

This chapter builds on the previous aalysis by focusing on system requirements. It defines the needs the proposed application must satisfy and guides understanding of user expectations.

The Unified Modeling Language (UML) is used to represent these requirements. The chapter presents UML overview, system actors, and key diagrams like use case and sequence diagrams.

## III.2 Definition of UML

We have chosen to use the Unified Modeling Language (UML) to model our system. UML is a rich visual modeling language that allows representing the different views of a system through various diagrams. It is mainly used in the architecture, design, and development of complex projects. In our case, UML will help us structure the processes, actors, and their interactions in the project for rectificatif documents

## III.3 Functional Analysis

Functional analysis describes the services that the system must provide to its users. These services are represented using Use Case Diagrams, which illustrate the interactions between users (actors) and the system.

### III.3.1 Functional Requirements

Functional requirements describe the essential functionalities that the system must provide in order to satisfy user requirements. Our application must allow different users to perform several operations related to rectification requests.Each functionality must be simple and clear to ensure good system usability.

### III.3.2 Non-Functional Requirements

Non-functional requirements, on the other hand, refer to criteria related to system quality, but which are not directly associated with its functionalities. They encompass aspects such as performance, security, reliability, and accessibility. For example, it is important that our system ensures the protection of sensitive data, is capable of quickly processing a large volume of reports, and is accessible on different platforms.

## III.4 Use Case Diagrams

The use case diagram serves to visually express the system's behavior in terms of actions and reactions, from each user's perspective. It defines the system, its functions (or uses), and its relationships with its environment.

### III.4.1 Identification of Actors

| Use case | Actors associated with | Details |
|---|---|---|
| Authentication | Student / Teacher / Administrator | -Login to the application |
| Request management | Student | -Add rectification request<br>-View requests<br>-Modify request<br>-Delete request |
| Request consultation | Teacher | -View students' requests |
| Rectification document management | Teacher | -View rectification document -<br>Manage rectification document history |
| Request validation | Administrator | -Validate rectification request<br>-Reject rectification request |
| Request monitoring | Administrator | -View all requests |
| User management | Administrator | -Manage users and access rights |

Table 3.1 - Identification of actors

## III.4.2 General Use Case Diagram

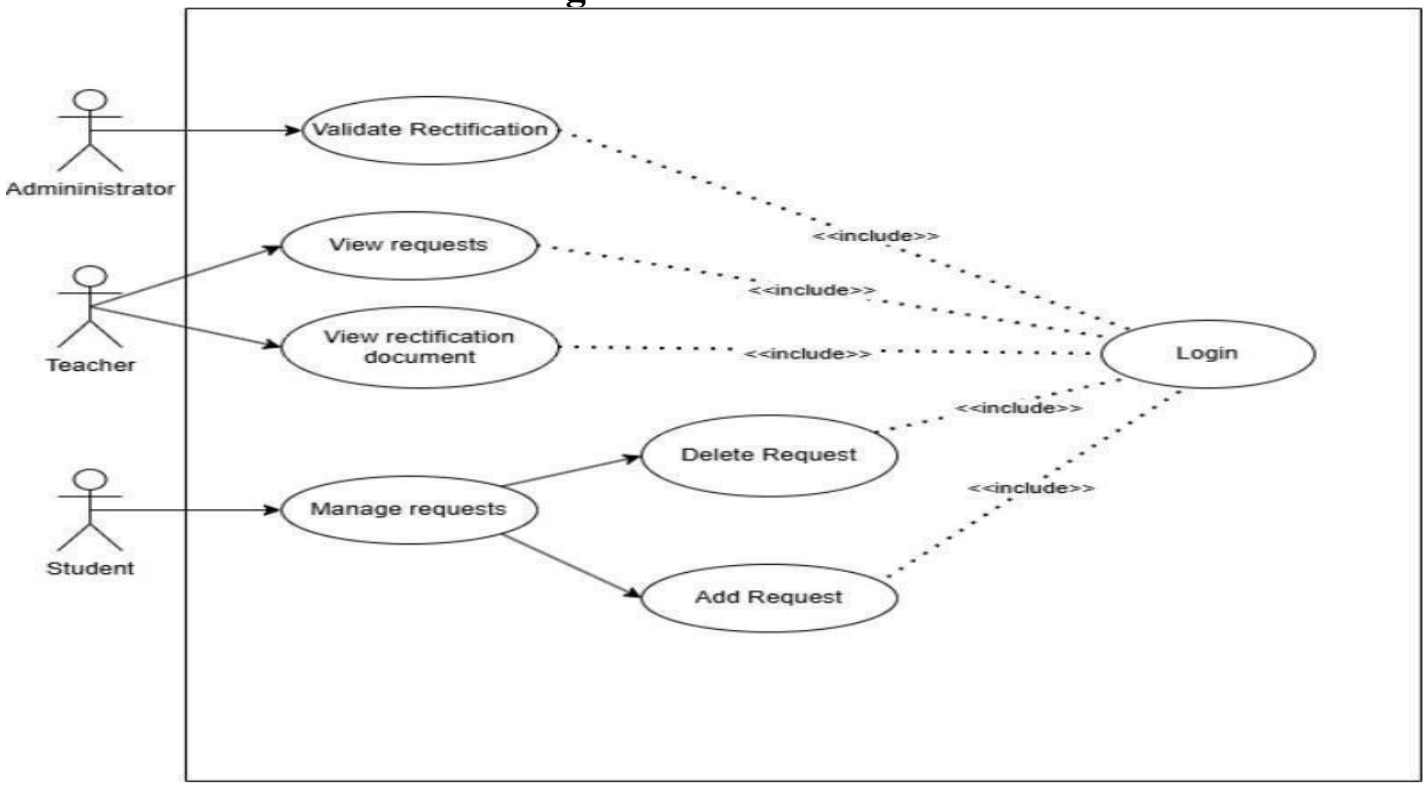



Figure 3.1- General use case diagram

### III.4.3 Administrator Use Case Diagram

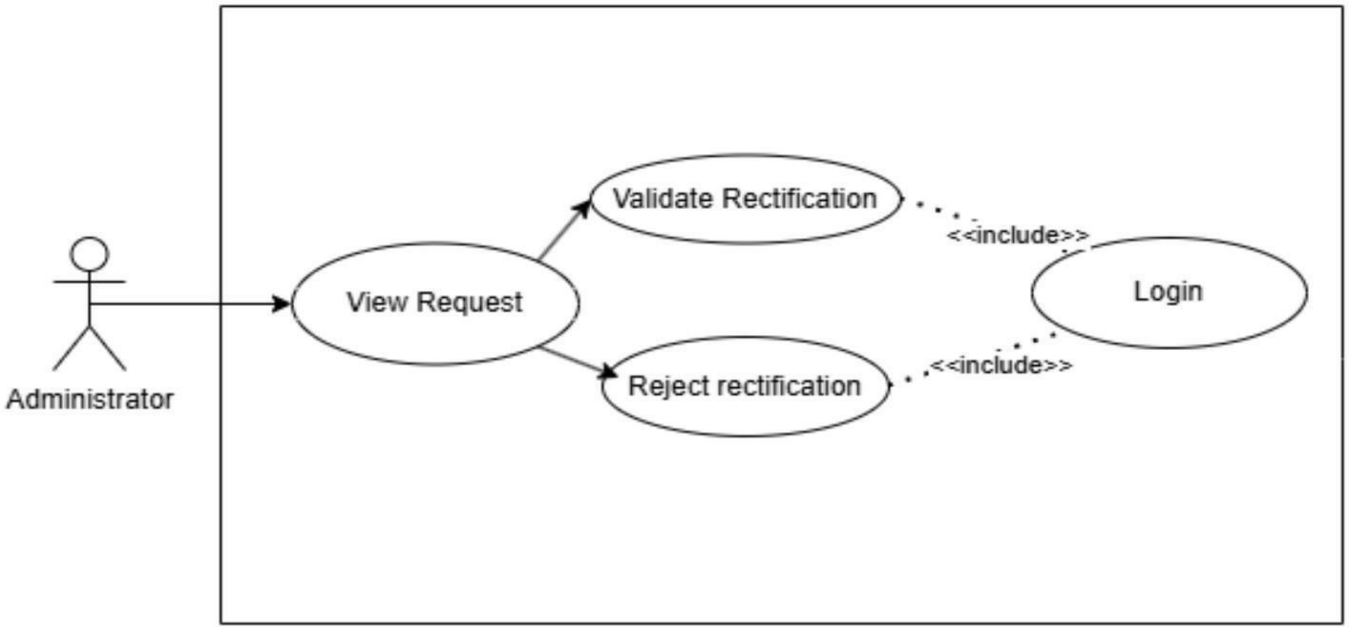



Figure 3. 2- Administrator use case diagram

### III.4.4 eacher Use Case Diagram :

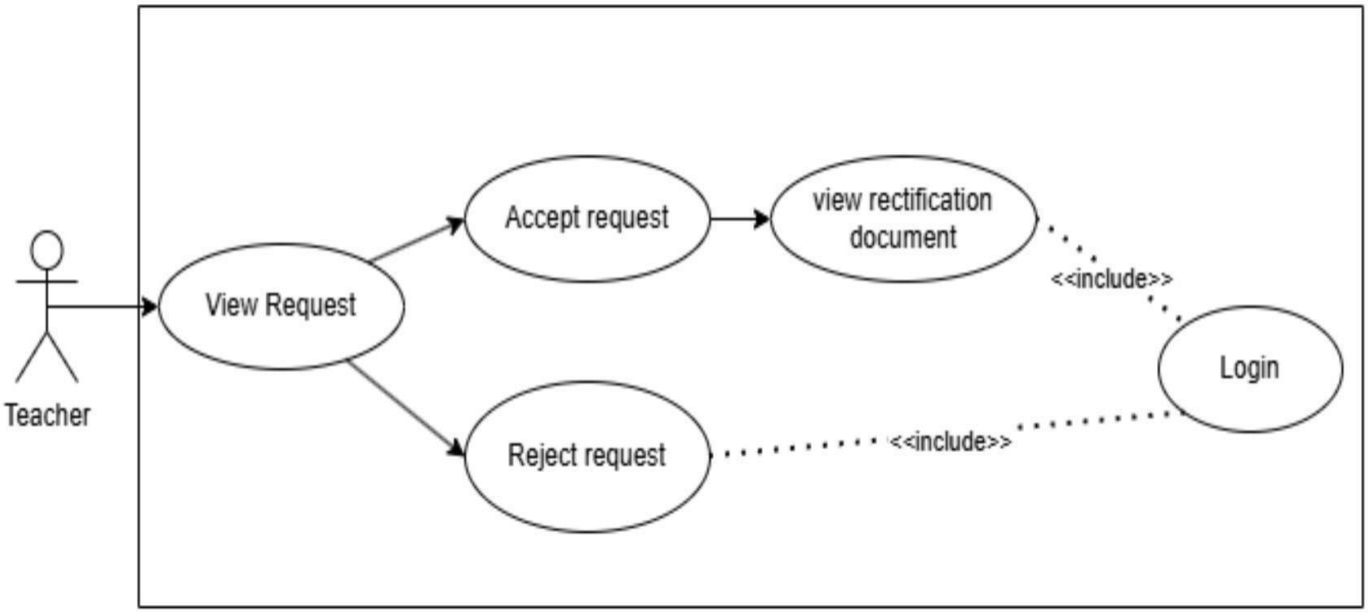



**Figure 3.3 - Teacher use case diagram**

### III.4.5 Student Use Case Diagram

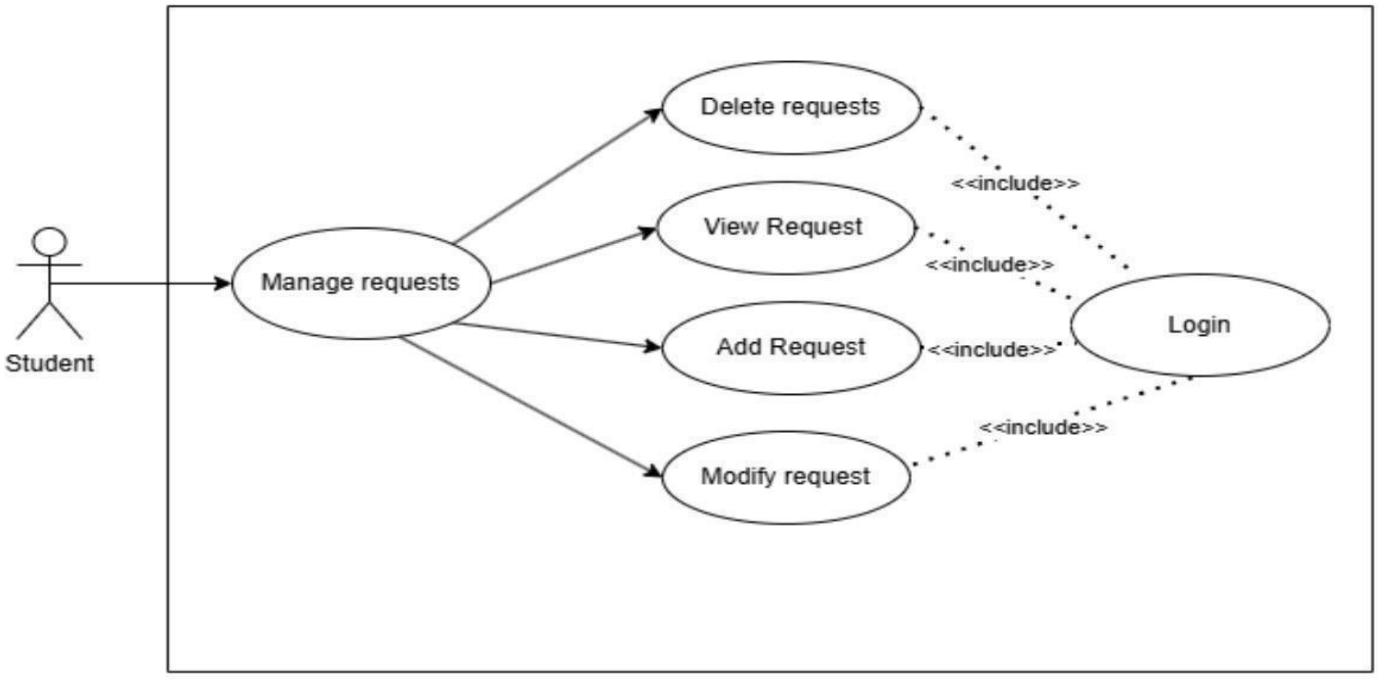



Figure 3.5 - Student use case diagram

## III.5 Text Description of Use Cases Use Cases

### III.5.1 Authentication

| Name | Authentication |
|---|---|
| Actor | Teacher/Administrator/ Student |
| Objective | Allow users to access their accounts |
| Pre-condition | The application is launched |
| Main flow | -The system displays the login form<br>-The user enters registration username and password<br>-The system verifies the information<br>-The system grants access |
| Alternative flow | If the information are incorrect, the system displays an error message |
| Post-condition | User is authenticated and redirected to the appropriate interface |

Table 3.2 -Use Cases Scenario: Authentication

### III.5.2 Add Request

| Name | Add Request |
|---|---|
| Actor | Student |
| Objective | Submit a rectification request |
| Pre-condition | Student is authenticated |
| Main flow | -The system displays modules<br>-The student chooses the module<br>-The system displays request form<br>-Student fills required data<br>-Student submits the request<br>-System stores request in database |
| Post-condition | Request is saved and pending for processing |

Table 3.3 -Use Cases Scenario: Add Request

### III.5.3 Accept Request

| Name | Accept Request |
|---|---|
| Actor | Teacher |
| Objective | Allow teachers to accept rectification requests |
| Pre-condition | Teacher is authenticated and requests are available |
| Main flow | -The system displays the pending requests<br>-The teacher selects a request<br>-The teacher checks the request information<br>-The teacher clicks accept request<br>-The system updates the request status to “Accepted” |
| Alternative flow | -If the request contains invalid information, the teacher rejects the request.<br>-The system displays an error message if the operation fails |
| Post-condition | -The request is approved successfully<br>-The request becomes available for administrator validation. |

Table 3.3 -Use Cases Scenario: Accept Request

### III.5.4 Validate Request

| Name | Validate Rectification |
|---|---|
| Actor | Administrator |
| Objective | Allow administrator to validate approved requests and generate the rectificatif document |
| Pre-condition | Administrator is authenticated and approved requests exist |
| Main flow | -The system displays approved requests.<br>-The administrator selects a request.<br>-The administrator validates the request.<br>-The system automatically generates the rectificatif document.<br>-The system stores the document in teacher history.<br>-The request status becomes “Validated”. |
| Alternative flow | -If the request is incorrect, the administrator rejects validation.<br>-The system displays an error message if validation fails. |
| Post-condition | -The request is validated successfully.<br>-The rectificatif document is generated and archived in history. |

Table 3.4 -Use Cases Scenario: Validate Request

## III.6 Sequence Diagram

Sequence diagrams illustrate how different objects interact with each other in a particular scenario. They show the chronological order of messages exchanged between actors and the system. These diagrams help understand the dynamic behavior of the system and how operations are executed step by step.

### III.6.1 Authentication Sequence Diagram

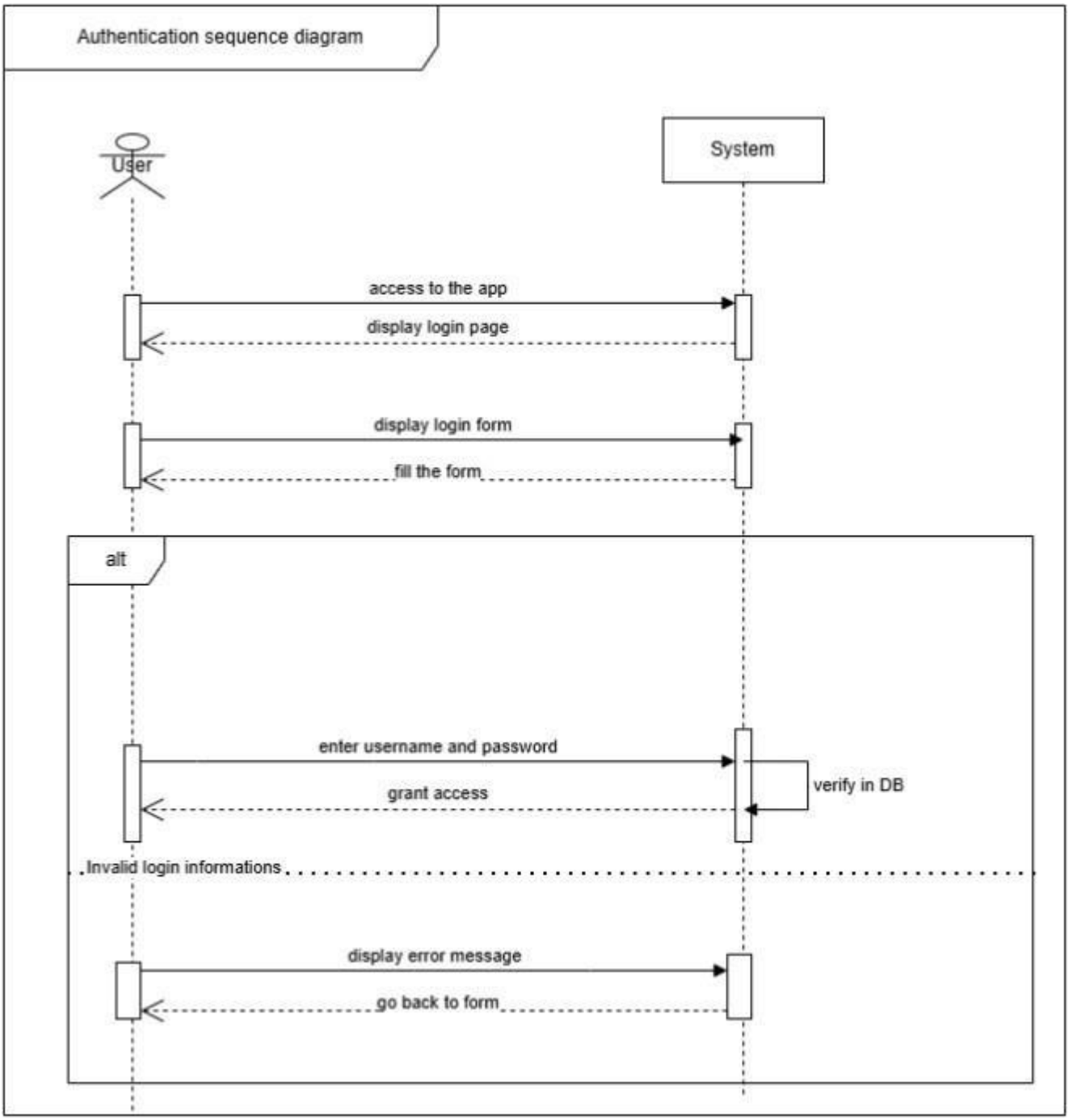



Figure 3.5 - Authentication sequence diagram

## III.6.2 Add Request Sequence Diagram

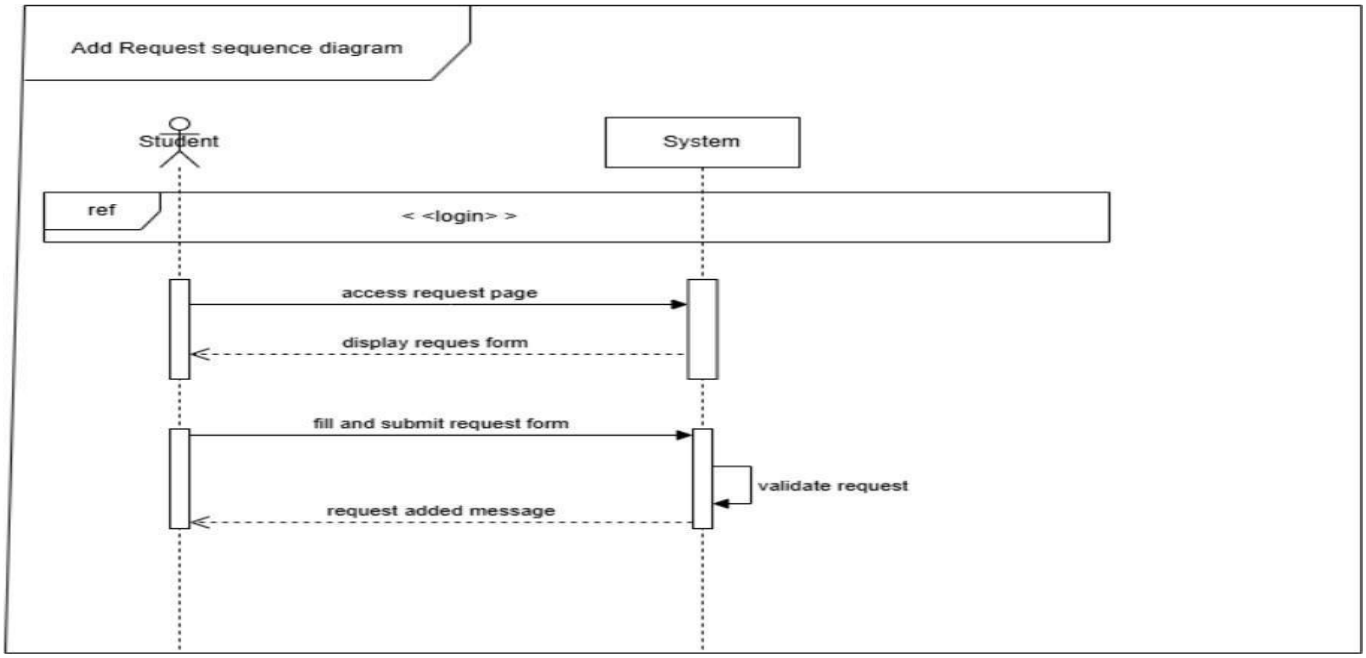



Figure 3.6 - Add request sequence diagram

## III.6.3 Accept Request Sequence Diagram

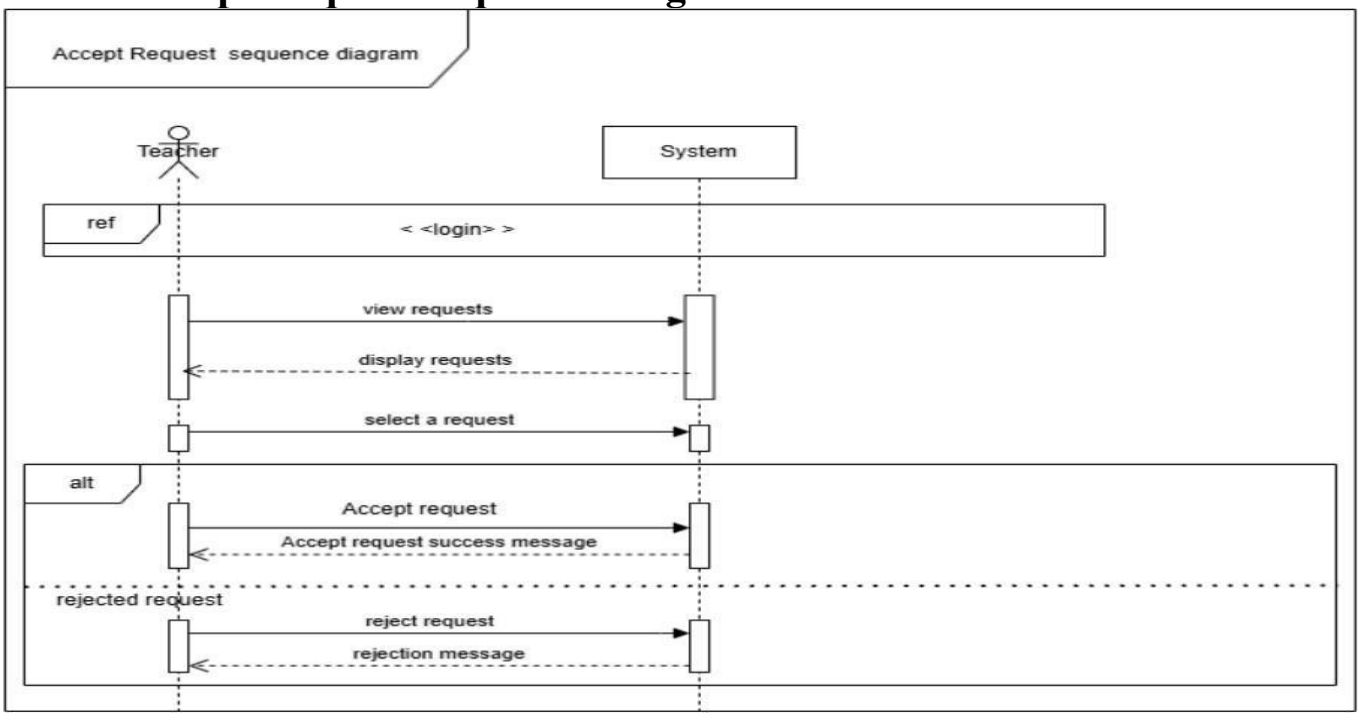



**Figure 3.7- Accept request sequence diagram**

### III.6.4 Validate Request Sequence Diagram

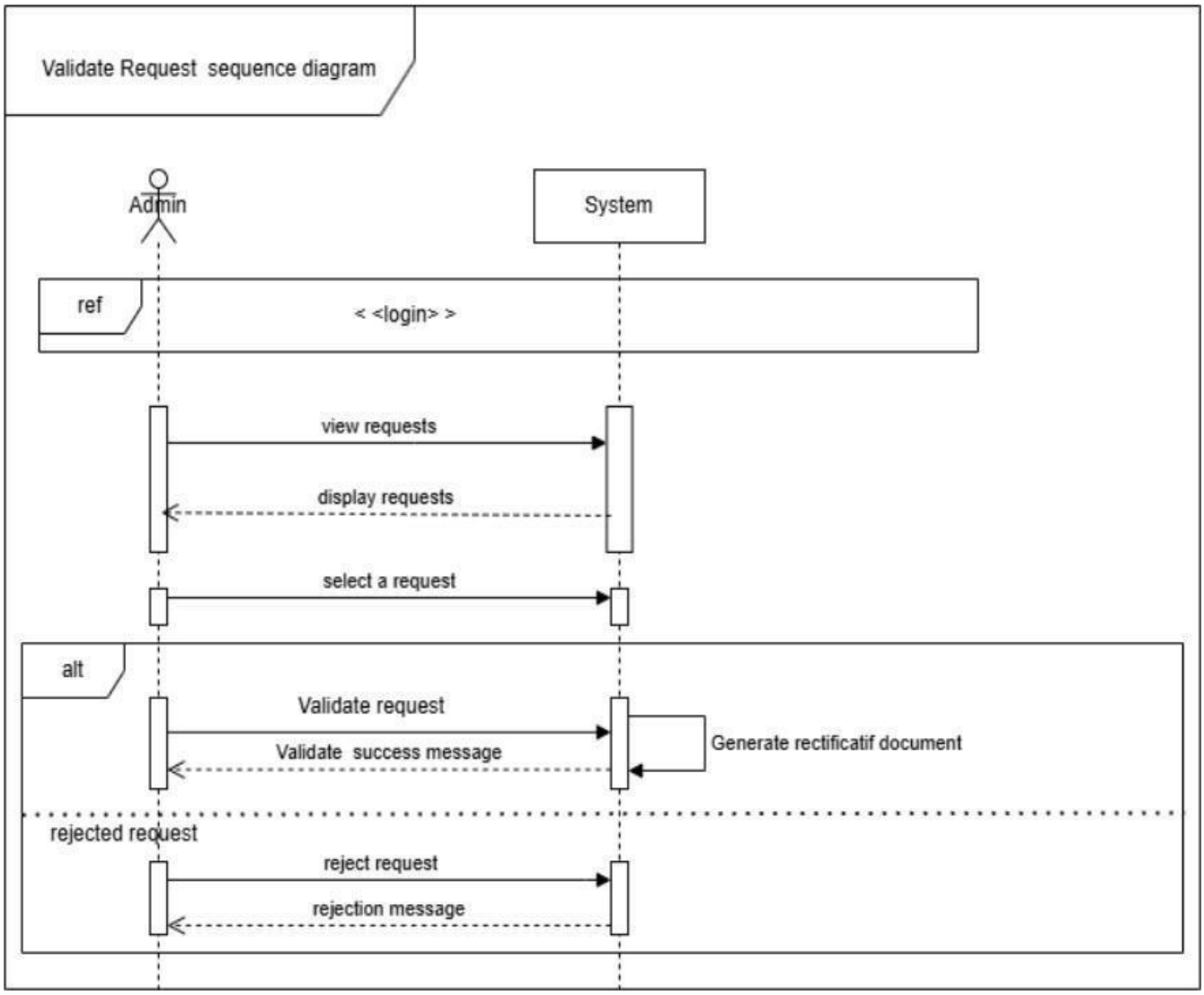



Figure 2.8- Validate request sequence diagram

## III.7 Conclusion

This chapter presented the system specification and requirement analysis of the proposed application, including UML modeling, functional and non-functional requirements, system actors, and use case and sequence diagrams. The next chapter will focus on system design, including class diagrams and the relational database model.

# Chapter 4: System Design

# Introduction:

After studying and analyzing the system requirements, we move on to the design phase. This step allows us to define the overall architecture of the application as well as the various interactions between the system's components. To model our application, we used UML through class diagrams and sequence diagrams.

## 1. Definition :

A class diagram is a UML diagram that represents the static structure of a system. It is used to describe the various classes, their attributes, their methods, and the relationships between them. It is an essential tool in object-oriented design because it provides an overview of the system's architecture prior to implementation. It also facilitates the understanding and modeling of the entities handled by the application. [12][13]

### 1.1 Class Diagram :

A class diagram is a structural diagram in UML (Unified Modeling Language) used in software engineering to represent the static structure of a system. It illustrates the system's classes, their internal structure (attributes), the operations they provide, and the relationships that exist between them.

#### 1.2.1 Purpose of Class Diagrams :

The main purpose of a class diagram is to provide a clear and visual representation of the system's static architecture. It helps to identify and organize system components, making it easier to understand, design, and document object-oriented systems.

#### 1.2.2 Example of a Class :

A class diagram element such as *MyClass* demonstrates the structure of a class, including its attributes and operations. Each operation may accept parameters and return a specific data type. For example, a method may return an integer, a float, or even a reference to another class depending on the system design.

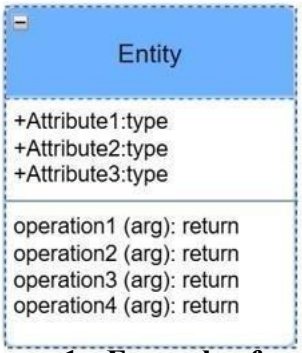



**Figure 1 - Exemple of a class**

### 1.2.3 Relationships between Classes 1

## Association

An association is a structural relationship that represents a connection between two or more classes. It indicates that objects of one class are linked to objects of another class and may communicate with each other in the system. [14]

## 2) Multiplicity

Multiplicity defines the number of instances of one class that can be associated with instances of another class. It specifies constraints such as one-to-one, one-to-many, or many-to-many relationships. [14]

## 3) Aggregation

Aggregation is a special type of association that represents a weak "whole–part" relationship. The part can exist independently of the whole, meaning its lifecycle is not strictly dependent on the parent class. [15]

## 4) Composition

Composition is a strong form of aggregation that represents a "whole–part" relationship where the lifetime of the part is strictly dependent on the whole. If the whole is destroyed, its parts are also destroyed. [15]

## 5) Dependency

A dependency is a relationship where one class relies on another class for its functionality. A change in the independent class may affect the dependent class. [14]

## 1. 2.4The Data Dictionary

We present below the Data Dictionary:

| Designation | Attribute Code | Type | Size |
|---|---|---|---|
| User identifier | id_user | INT | 11 |
| Student identifier | id_student | INT | 11 |
| Teacher identifier | id_teacher | INT | 11 |
| Administrator identifier | id_admin | INT | 11 |
| Reclamation identifier | id_reclamation | INT | 11 |
| Note identifier | id_note | INT | 11 |
| First name | first_name | VARCHAR | 255 |
| Last name | last_name | VARCHAR | 255 |

| | | | |
|---|---|---|---|
| Email | email | VARCHAR | 255 |

20

| Designation | Attribute Code | Type | Size |
|---|---|---|---|
| Password | password | VARCHAR | 255 |
| Matricule | matricule | VARCHAR | 50 |
| Module name | module | VARCHAR | 255 |
| Old mark | old_mark | FLOAT | 10 |
| New mark | new_mark | FLOAT | 10 |
| Status | status | VARCHAR | 50 |
| Message content | message | TEXT | — |
| Creation date | created_at | DATE | — |

***Table** 2 – **Data Dictionary of the PV Rectification System***

## 2. The Proposed Class Diagram

The following UML class diagram illustrates the complete static structure of the Exam PV Rectification System. It shows all classes, their attributes, their methods, and the relationships that bind them together.

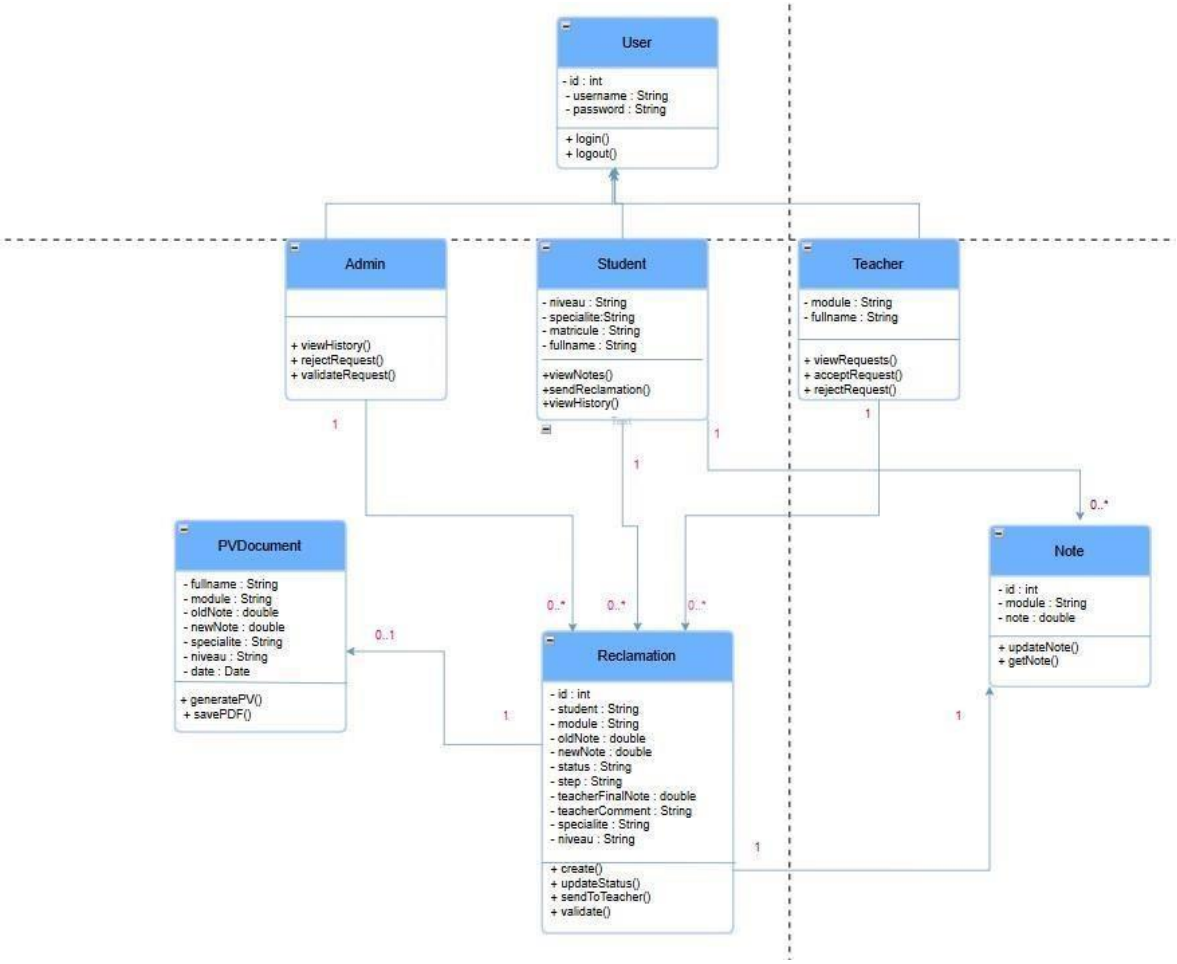



***Figure** 2 – UML Class Diagram of the Exam PV Rectification System*

## 3. Management Rules :

The class diagram was developed according to the following management rules:

- A student can have one or more grades.
- Each grade belongs to one and only one student.
- A student can submit one or more reclamations.
- Each reclamation is submitted by one and only one student.
- A reclamation is associated with one grade.
- A grade can be the subject of one or more reclamations.
- A teacher can review and process multiple reclamations.
- Each reclamation is assigned to one teacher.
- A reclamation can be accepted, rejected, or remain pending.
- may result in a grade update.

## IV.3 The Relational Model:

By applying the transformation rules from the object-oriented model to the relational model based on the class diagram, the following relations are obtained:

- Student (id_student, first_name, last_name, matricule, niveau, specialite, password).
- Teacher (id_teacher, first_name, last_name, module, password).
- Administrator (id_admin, first_name, last_name, password).
- Note (id_note, module, mark, #id_student).
- Reclamation (id_reclamation, message, status, submission_date, teacher_comment, #id_student, #id_teacher, #id_note).
- PVDocument (id_pv, old_mark, new_mark, generation_date, #id_reclamation)

## IV.7 Detailed Sequence Diagrams:

A sequence diagram is a UML behavioral diagram that represents the interactions between objects or actors in a system according to a chronological sequence. It illustrates how messages are exchanged to accomplish a specific functionality and helps visualize the dynamic behavior of the system. Sequence diagrams are widely used to model system scenarios and describe the flow of operations between different components of an application [14][15].

The following diagrams illustrate the main functionalities of the proposed system

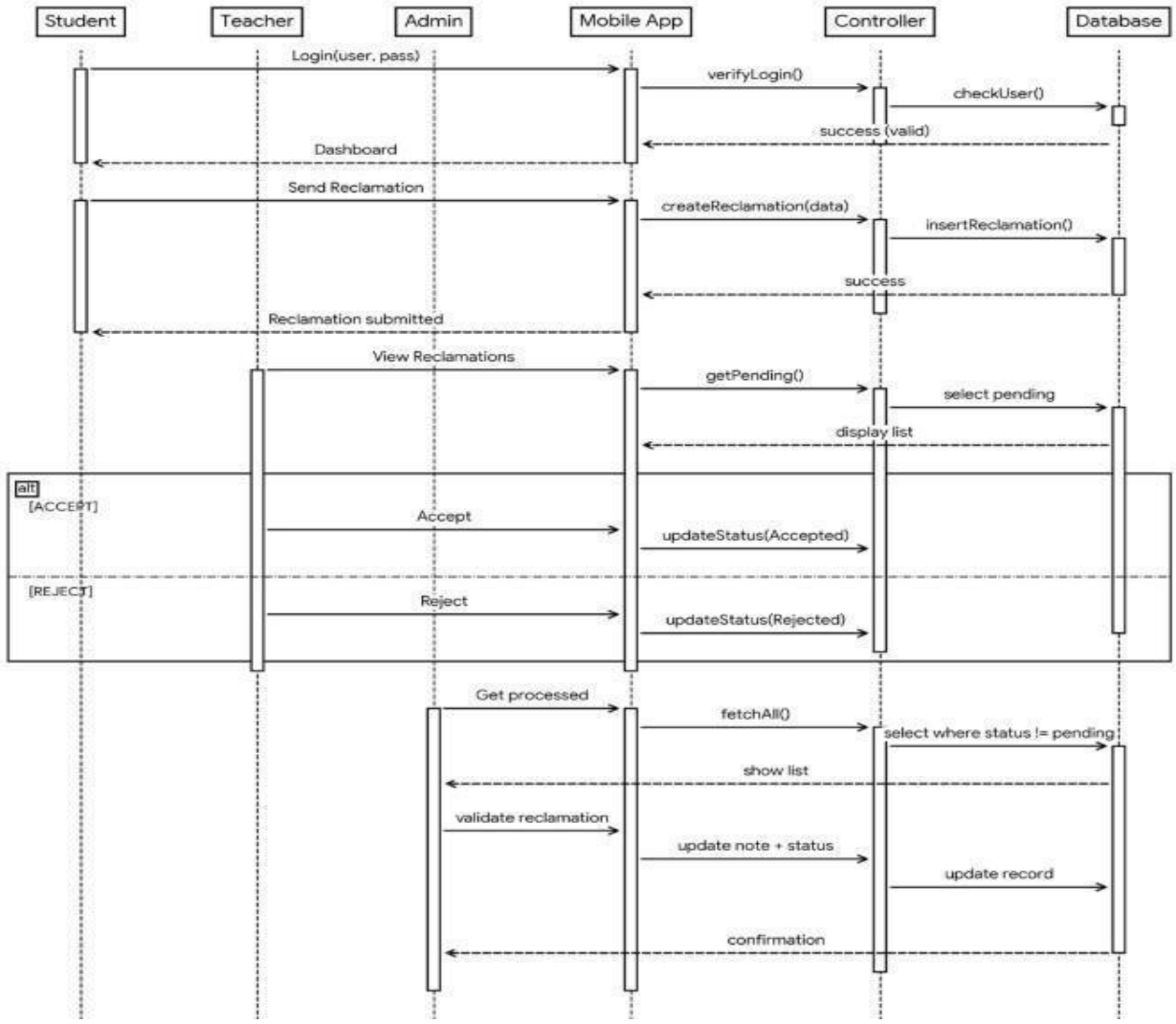



*Figure 3 - Detailed sequence diagram*

## IV.8 Conclusion :

In this chapter, we presented the design of the proposed system through the class diagram, the relational model, and the sequence diagrams. This phase provides the foundation for the implementation presented in the next chapter.

# Chapter 5: Implementation

# Introduction:

This chapter presents the implementation phase of the PV Rectificatif management application. After establishing the theoretical foundations in Chapter 1, analyzing the existing system in Chapter 2, specifying the requirements in Chapter 3, and designing the system architecture in Chapter 4, we now describe the concrete realization of the collaborative mobile application. This chapter details the development environment, the technologies and tools used, and the presentation of the application interfaces.

## V.1 Development Environment :

The development environment constitutes the set of hardware and software tools used to build, test, and deploy the PV Rectificatif application. The choice of appropriate tools is essential to ensure code quality, developer productivity, and application reliability.

### V.1.1 Hardware Environment :

The application was developed on the following hardware configuration:

| Component | Specification |
|---|---|
| **Processor** | Intel Core i5 / AMD Ryzen 5 or equivalent |
| **RAM** | 8 GB minimum |
| **Storage** | 256 GB SSD |
| **Operating System** | Windows 10/11 (64-bit) |
| Test Device | Device Android smartphone (API level 26+) |

### V.1.2 Software Environment :

The following software tools were used throughout the development process:

| Tool | Description |
|---|---|
| **Android Studio** | Official IDE for Android development (Hedgehog 2023.1) |
| **VS Code** | Code editor used for backend and API scripting |
| **Python 3.x + Flask** | Backend framework for the REST API server |
| **MySQL** | Relational database management system |
| **Postman** | API testing and validation tool |
| **Git / GitHub** | Version control and collaborative source code management |
| **Draw.io** | UML diagram design tool (use cases, class, sequence) |

## V.2 Technologies and Programming Languages :

### V.2.1 Kotlin (Android)

The mobile application was developed using Kotlin, the official programming language recommended by Google for Android development. Kotlin offers full interoperability with

Java, concise syntax, null safety features, and strong support for modern Android architecture components such as ViewModel, LiveData, and Retrofit.

## V2.2 Python / Flask (Backend)

The server-side logic was implemented using Python with the Flask microframework. Flask was chosen for its lightweight architecture, ease of use, and flexibility in building RESTful APIs. The backend handles authentication, data processing, request management, and PDF generation.

## V2.3 MySQL (Database)

The application uses MySQL as its relational database management system. The database stores all user accounts (students, teachers, administrators), rectification requests, correction documents, and validation records. The relational schema is directly derived from the class diagram designed in Chapter 4.

## V2.4 REST API Architecture

Communication between the Android mobile application and the backend server is implemented through a REST API using HTTP methods (GET, POST, PUT). Data is exchanged in JSON format, ensuring a clear separation between the frontend and backend layers. Retrofit library was used on the Android side to handle API calls asynchronously.

# V3 Application Interfaces

## Authentication Interface Login

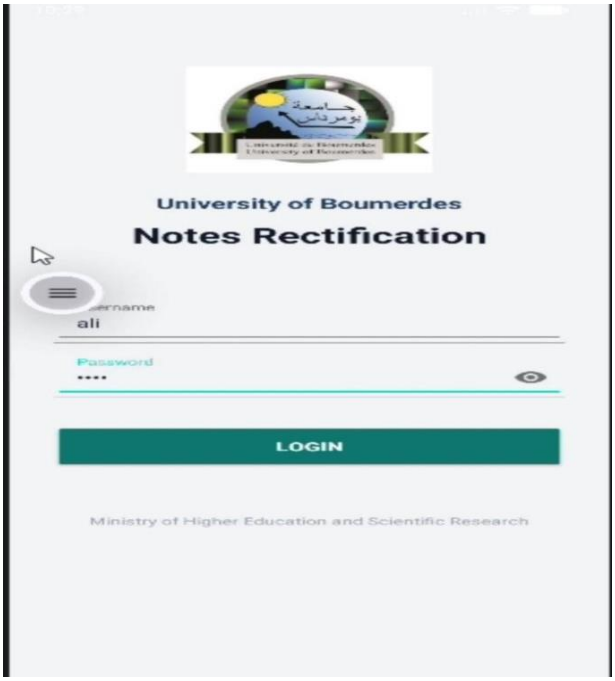



Figure V.1 – Login Interface of the PV App

## V3.2 Student Dashboard Figure :

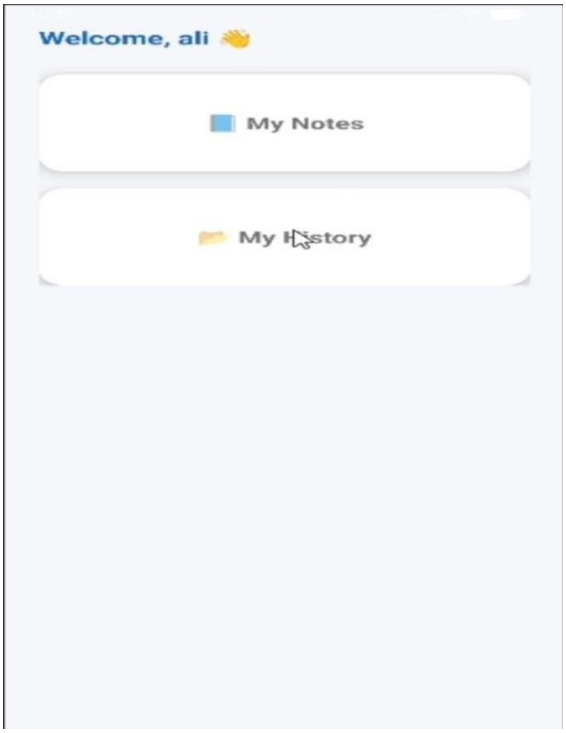
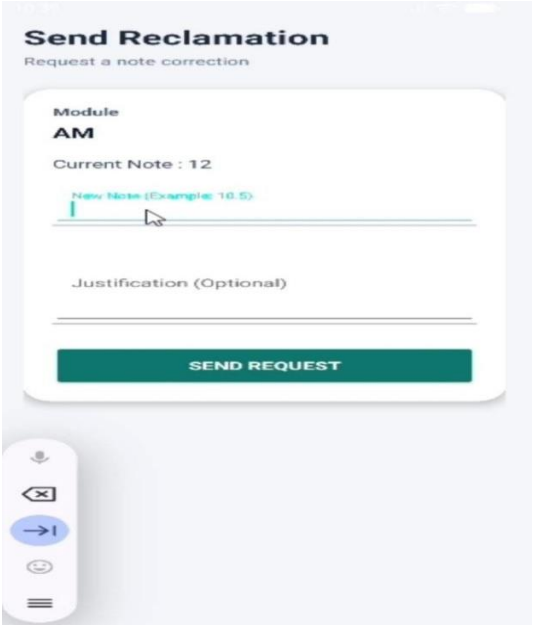



Figure V.2 - Student Dashboard Figure

**V3.3** Grade Rectification Request Interface Grades and Status List Figure




V.3 – Rectification Request Form

V3.4 Send to Teacher Interface

Figure




Figure V.4 – Send to Teacher Confirmation Screen

V3.5 Grades and Status List

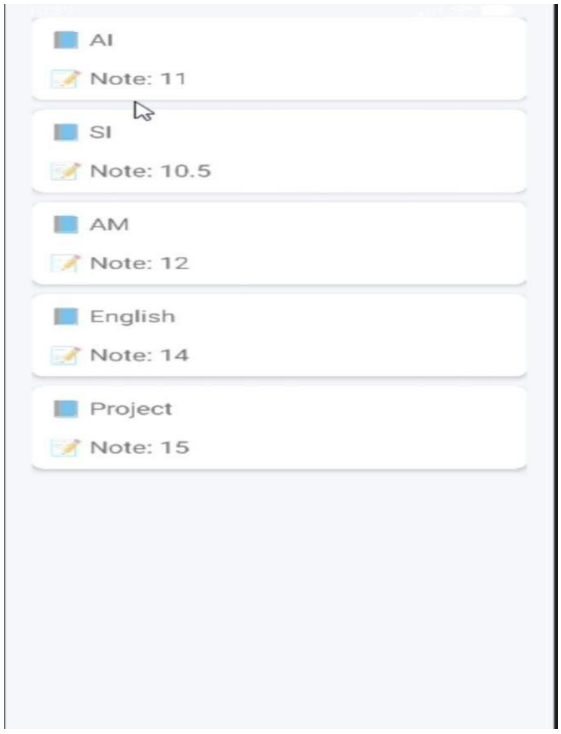



Figure V.5 – Grades and Rectification Status List

V3.6 Request History Interface

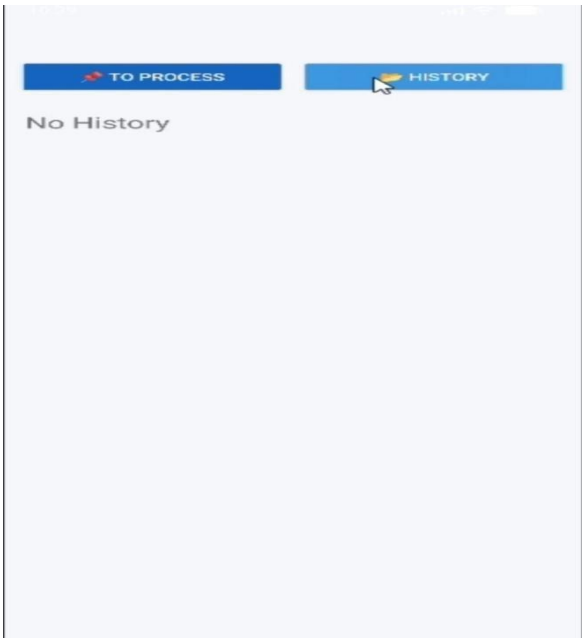



Figure V.6 – Request History Interface

V.7 – Generated PV Rectification PDF Document

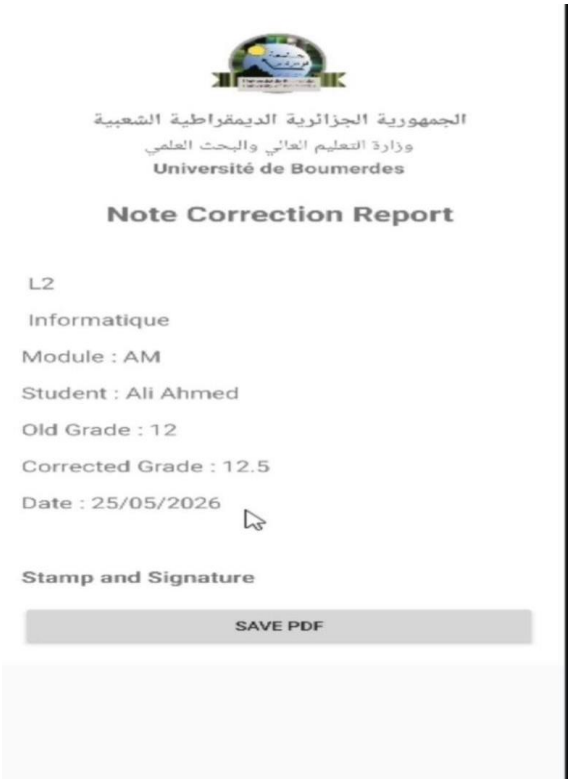



Figure V.7 – Generated PV Rectification PDF Document

## Conclusion

This chapter presented the implementation of the PV Rectificatif application, covering the development environment, the technologies used (Kotlin, Flask, MySQL, REST API), and the main application interfaces. The application successfully addresses the problems identified in Chapter 2, providing a digitalized, collaborative, and traceable solution for managing grade correction reports in the Algerian university context.

# General Conclusion

The work presented in this dissertation aimed to design and develop a collaborative mobile application for the management of grade correction reports (PV Rectificatifs) at the University of M'hamed Bougara Boumerdes. The project was motivated by the limitations of the current manual system, which relies on paper documents and informal communications, leading to delays, errors, and lack of traceability. Through five structured chapters, we progressively built the theoretical, analytical, and practical foundations of our solution. Chapter 1 introduced the concepts of digital collaboration and collaborative platforms.

Chapter 2 examined the specific context of Algerian universities and identified the key problems.

Chapter 3 formalized the system requirements through UML diagrams.

Chapter 4 provided the complete conceptual design. Finally,

Chapter 5 presented the concrete implementation and the application interfaces. The resulting application — PV App — is an Android mobile platform that enables students to submit grade rectification requests, teachers to review and validate corrections, and administrators to approve final documents. The system automatically generates official PV documents in PDF format bearing the University of Boumerdes letterhead, fully replacing the need for physical paperwork. As future perspectives, we envision: a dedicated administrative web interface, integration with the university's existing student information systems, digital signature support, push notification capabilities, and multi-language support including Arabic.

# References


[1] *Workforce Experience, Digital Collaboration Trends, 2025.* https://workforceexperience.hp.com/blog/what-is-digital-collaboration/

[2] *ScienceDirect, Collaborative Production in Digital Teams, 2025.* https://www.sciencedirect.com/topics/computer-science/collaborative-production

[3] *Chen, P., & Nunamaker, J. F., Collaborative Systems and Platforms, 2003.* https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1164&context=jitim

[4] *Google, Project Aristotle: Understanding Team Effectiveness, 2015.* https://rework.withgoogle.com/intl/en/guides/understanding-team-effectiveness

[5] *Benfoughal, A., & Badissi, R. (2024). Digital Transformation and E-learning Integration in Algerian Higher Education.* ASJP. https://asjp.cerist.dz/en/article/253637

[6] *Dahbi, S., & Bouhali, M. (2023). Activating Digital Education in Algerian Universities: Reality and Challenges.* ASJP. https://asjp.cerist.dz/en/article/258554

[7] *Ameur, N., et al. (2024). The Mechanism of Digital Transformation in Higher Education: University of M'sila Case Study.* ASJP. https://asjp.cerist.dz/en/article/265371

[8] *Booch, G., Rumbaugh, J., & Jacobson, I. (2005).* The Unified Modeling Language User Guide (2nd ed.). Addison-Wesley Professional.

[9] *Merise / UML – Passage du diagramme de classes au modèle relationnel.* Cours de modélisation, Université de Paris 13, IUT Villetaneuse, 2013.

[10] *Fowler, M. (2003).* UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley Professional.

[11] *Google. (2023).* Kotlin Programming Language — Official Documentation. https://kotlinlang.org/docs/

[12] Booch G., Rumbaugh J., Jacobson I., *The Unified Modeling Language User Guide*, 2nd Edition, Addison-Wesley, 2005.

[13] Object Management Group (OMG), *Unified Modeling Language (UML) Specification*, Version 2.5.1, 2017.

[14] Object Management Group (OMG), *Unified Modeling Language (UML) Specification*, Version 2.5.1, 2017.

[15] Booch G., Rumbaugh J., Jacobson I., *The Unified Modeling Language User Guide*, 2nd Edition, Addison-Wesley, 2005.