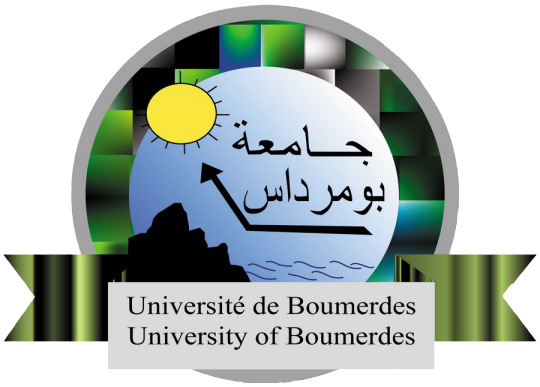
People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of M'hamed Bougara-Boumérdes
Computer science faculty



Graduation Thesis for **the degree of Bachelor**

In : Computer science

Specialization : Computer systems

By : KISSOUM Celia , METRITER Lea , GOURI Rahma , ISLI Nourelhouda

Title

Online Exam Management Platform.

Graduated , On **03/06/2026**, in front of the following examination committee:

M.	Kedjar Saadia	Professor	At UMBB	Supervisor
M.	Hamadouche Samia	Professor	At UMBB	Examiner

We dedicate this work: To our beloved parents, for their unconditional love, endless support, and sacrifices throughout our academic journey. To our families, for their encouragement, patience, and constant presence. To our teammates, for their collaboration, dedication, and the valuable moments we shared during this project. Finally, to everyone who contributed, directly or indirectly, to the successful completion of this work.

Acknowledgement

We would like to express our deepest gratitude to our supervisor for their invaluable guidance, constructive feedback, and continuous support throughout the realization of this project. We also extend our sincere thanks to all the members of the teaching staff for the quality of their instruction and their dedication during our academic training. Finally, we would like to acknowledge all those who have contributed, directly or indirectly, to the successful completion of this work.

Abstract

With the rapid advancement of digital technologies, online examination systems have become essential tools in the field of education. They facilitate exam organization, improve accessibility, and simplify the management of results. In this context, this Bachelor's thesis focuses on the design and development of an online examination platform called UniExam.

The main objective of this platform is to enable students to take examinations remotely and view their results, while providing professors with tools to create and manage exams, add questions, and evaluate student performance. In addition, administrators are provided with functionalities to manage users and maintain system data efficiently.

The system was designed using the Unified Modeling Language (UML) to model its structural and functional aspects. The implementation was carried out using HTML, CSS, JavaScript, and PHP, with MySQL as the database management system and WAMP Server as the local development environment.

The results obtained demonstrate that UniExam is a reliable, user-friendly, and efficient solution for managing university online examinations. The platform contributes to the digital transformation of educational institutions by improving the accessibility, flexibility, and efficiency of the examination process.

Keywords: Online Examination System, E-learning, UML, HTML, CSS, JavaScript, PHP, MySQL, UniExam.

الملخص

مع التطور السريع للتقنيات الرقمية، أصبحت أنظمة الامتحانات الإلكترونية من الأدوات الأساسية في مجال التعليم، حيث تساهم في تسهيل تنظيم الامتحانات وتحسين إمكانية الوصول إليها وتبسيط إدارة النتائج. في هذا الإطار، يهدف مشروع التخرج هذا إلى تصميم وتطوير منصة للامتحانات عبر الإنترنت تحمل اسم UniExam.

تهدف هذه المنصة إلى تمكين الطلبة من اجتياز الامتحانات عن بعد والاطلاع على نتائجهم، كما توفر للأساتذة إمكانية إنشاء الامتحانات وإدارتها وإضافة الأسئلة وتصحيحها، بالإضافة إلى تمكين المسؤولين من إدارة المستخدمين والبيانات الخاصة بالنظام.

تم تصميم النظام بالاعتماد على لغة النمذجة الموحدة (UML) **من أجل نمذجة مختلف الجوانب الوظيفية والبنيوية للتطبيق. أما عملية التطوير فقد تم إنجازها باستخدام** HTML **و**CSS **و**JavaScript **و**PHP**، مع استخدام** MySQL **كنظام لإدارة قواعد البيانات، و**WAMP Server **كبيئة تطوير محلية.**

أظهرت النتائج المتحصل عليها أن منصة UniExam **تمثل حلاً موثوقاً وسهل الاستخدام وفعالاً لإدارة الامتحانات الجامعية عبر الإنترنت، مما يساهم في دعم التحول الرقمي داخل المؤسسات التعليمية.**

الكلمات المفتاحية: الامتحانات الإلكترونية، التعليم الإلكتروني، UML، HTML، CSS، JavaScript، PHP، MySQL، UniExam.

Contents

Abstract i

List of Figures iv

List of Tables v

General Introduction 1

1 Presentation of the Online Evaluation System 2
- 1.1 Introduction . . . 2
- 1.2 Traditional exams . . . 2
  - 1.2.1 Definition . . . 2
  - 1.2.2 Different Types of Exams . . . 2
  - 1.2.3 Historical Origins of Exams . . . 2
- 1.3 Online Exams . . . 3
  - 1.3.1 Definition . . . 3
  - 1.3.2 History of Online Exams . . . 3
- 1.4 Online Formative/Summative Evaluation System . . . 3
  - 1.4.1 Summative Assessment . . . 3
  - 1.4.2 Formative Assessment . . . 3
  - 1.4.3 Comparison Between Formative and Summative Assessment . . . 3
  - 1.4.4 Examples of Online Formative and Summative Assessment Systems . . . 4
- 1.5 Advantages and Challenges of Online Examinations . . . 6
  - 1.5.1 Advantages . . . 6
  - 1.5.2 Challenges . . . 6
- 1.6 Conclusion . . . 6

2 System Analysis And Conception 7
- 2.1 Introduction . . . 7
- 2.2 Needs Analysis . . . 7
  - 2.2.1 Functional Needs . . . 7
  - 2.2.2 Non-functional Needs . . . 8
- 2.3 Actors Identification . . . 8
- 2.4 Use Case Diagram . . . 8
- 2.5 Sequence Diagram . . . 9
  - 2.5.1 Sequence Diagram: Authenticate . . . 10
  - 2.5.2 Sequence Diagram: Manage Notifications . . . 11
  - 2.5.3 Sequence Diagram: Administrator . . . 12
    - 2.5.3.1 Sequence Diagram: Manage Users . . . 12
    - 2.5.3.2 Sequence Diagram: Schedule Planning . . . 13
  - 2.5.4 Sequence Diagram: Professor . . . 14
    - 2.5.4.1 Sequence Diagram: Create Exam . . . 14
    - 2.5.4.2 Sequence Diagram: Consult Exam . . . 15
    - 2.5.4.3 Sequence Diagram: Modify Exam . . . 15
  - 2.5.5 Sequence Diagram: Student . . . 16

2.5.6 Use Cases Description Tables . . . 17
2.6 Data Modeling (Class Diagram) . . . 19
2.6.1 Data Dictionary . . . 19
2.6.2 Business Rules . . . 20
2.6.3 Class Diagram: . . . 22
2.6.4 Database Relational Model . . . 22
2.7 Rules of Passage: Class Diagram → Relational Model . . . 23
2.8 Conclusion . . . 26

3 Implementation 27
3.1 Introduction . . . 27
3.2 Software Environment . . . 27
3.2.1 Development Tools . . . 27
3.2.2 Development Languages . . . 27
3.3 Choice of DBMS (Database Management System) . . . 28
3.4 Deployment Diagram . . . 29
3.5 Graphic Interfaces Presentation . . . 30
3.5.1 Authentication Interface . . . 30
3.5.2 Notifications Managing Interfaces . . . 30
3.5.3 Admin Space . . . 31
3.5.4 Professor Space . . . 32
3.5.5 Student Taking Exam Interface . . . 34
3.6 Conclusion . . . 34

General Conclusion 35

Webography 36

List of Figures

1.1 Trigonometry online test on moodle . . . 4
1.2 Login page of the platform . . . 4
1.3 An Online Exam on GoogleForms . . . 5
1.4 Main page of the platform. . . . 5
1.5 ProctorU landing page for remote exam supervision. . . . 5
1.6 Functional overview of the TestWe online assessment solution. . . . 6

2.1 General Use Diagram . . . 9
2.2 Sequence Diagram For Authentication . . . 10
2.3 Sequence Diagram for Notifications Management . . . 11
2.4 Sequence Diagram for Users Management . . . 12
2.5 Sequence Diagram for planning scheduling . . . 13
2.6 Sequence Diagram for Exam Creation . . . 14
2.7 Sequence Diagram for Exams Consultation . . . 15
2.8 Sequence Diagram for Exam Editing . . . 15
2.9 Sequence Diagram for taking an Exam . . . 16
2.10 Class Diagram . . . 22

3.1 Deployment Diagram of the Online Examination System. . . . 29
3.2 Authentication Interface. . . . 30
3.3 Notifications Managing Interfaces. . . . 31
3.4 Exam planning Scheduling Interface. . . . 32
3.5 Users Managing Interface. . . . 32
3.6 Exam Consultation Interface . . . 33
3.7 Exam Creation Interface . . . 33
3.8 Exam Editing Interface. . . . 34
3.9 Student Taking Exam Interface. . . . 34

List of Tables

1.1 Comparison Between Formative and Summative Assessment 3

2.1 Use Case Description: Authenticate 10
2.2 Use Case Description: Manage Notifications 11
2.3 Use Case Description: Manage Users 13
2.4 Use Case Description: Schedule Planning 13
2.5 Use Case Description: Create Exam 14
2.6 Use Case Description: Consult / Modify Exam 16
2.7 Use Case Description: Take Exam 17
2.8 Use Case Description: View Dashboard 17
2.9 Use Case Description: Manage Academic Structure 17
2.10 Use Case Description: View Planning 17
2.11 Use Case Description: View Student Submissions 18
2.12 Use Case Description: Grade Exam Submissions 18
2.13 Use Case Description: Export / Upload Results 18
2.14 Use Case Description: View Available Exams 18
2.15 Use Case Description: View Results 19
2.16 Data Dictionary — UniExam UMBB 19
2.17 Business Rules — UniExam UMBB 20
2.18 Rule 1 — Class to Table Transformation 24
2.19 Rule 2 — Inheritance Transformation 24
2.20 Rule 3 — One-to-Many Transformation 25
2.21 Rule 4 — Many-to-Many Transformation 25
2.22 Rule 5 — Multiple Associations Transformation 25
2.23 Summary — Final Relational Model (20 Tables) 26

General Introduction

- **Context and Motivation:** In recent years, the Internet has become an essential part of daily life, affecting several fields, especially education. Thanks to the development of digital technologies, distance learning has widely expanded, allowing learners to access courses and take exams online without geographical constraints.

  In this context, online evaluation systems have gained increasing importance as they provide effective solutions for managing exams remotely. These systems offer several advantages, such as saving time, providing flexibility, and simplifying the management of examinations and results.

  The main objective of this project is to design and develop an online examination system for schools and universities. This system enables students to take exams online and view their results, while allowing teachers to manage exams, questions, and grades. In addition, administrators can manage users and maintain system data efficiently.

  To achieve this objective, UML was used for system modeling and design. The implementation is based on HTML, CSS, and JavaScript for the front-end, and PHP with MySQL for the back-end. The development environment used is WAMP Server.

  This document is organized into three chapters. The first chapter presents the general concepts related to online evaluation systems. The second chapter focuses on the system design using UML diagrams. Finally, the third chapter describes the implementation process and the technologies used.

- **Problem Statement:** Traditional examination management in schools and universities often relies on paper-based processes or limited digital tools. These methods can be time-consuming, difficult to manage, and prone to errors in exam preparation, distribution, correction, and result publication. In addition, ensuring academic integrity during online examinations remains a major challenge, as students may attempt to cheat using unauthorized resources or external assistance. With the growing demand for distance learning and digital education services, educational institutions need a reliable solution that allows examinations to be conducted efficiently while maintaining fairness, security, and accessibility. Therefore, there is a need for an online examination system that simplifies exam management, automates evaluation processes, and includes mechanisms to reduce cheating during examinations.

- **Proposed Solution:** To address these challenges, we propose the development of UniExam, a web-based online examination platform designed for schools and universities. The platform enables students to take exams online and view their results, while allowing professors to create and manage examinations, questions, and grades. Administrators can manage users and oversee the overall operation of the system. The proposed solution integrates secure authentication, role-based access control, automated correction of objective questions, exam time management through a timer, and result publication features. In addition, the platform includes an anti-cheating monitoring system that helps detect suspicious behavior during examinations and supports academic integrity. Developed using HTML, CSS, JavaScript, PHP, and MySQL, with WAMP Server as the development environment, UniExam provides a user-friendly, efficient, secure, and scalable solution for managing online examinations.

Chapter 1

Presentation of the Online Evaluation System

1.1 Introduction

Online learning platforms have enabled many educational institutions to rely on digital (online) exams to organize assessments. They have become widely adopted due to the time and cost savings they offer. In this context, we will discuss the definition of traditional and electronic exams, understand their characteristics, and present examples of programs currently in use.

1.2 Traditional exams

1.2.1 Definition

A traditional exam is a structured set of questions related to a specific subject or field, aimed at evaluating the student's level of understanding and the skills acquired during their training. It is generally organized at the end of a learning period (semester, academic year, competitive exam, etc.). A traditional exam can be written, oral, or practical, depending on the nature of the subject. In our work, we will mainly focus on written exams. These are graded through a scoring system assigned by the teacher, who corrects the papers and determines the final grade. Students take the exam under the supervision of a teacher and must be physically present in the institution or examination room.

1.2.2 Different Types of Exams

Exams take various forms and include several types. We will mention a few examples:

• Among these forms,**written exams** are commonly used. They can appear in different formats, such as multiple-choice questions (MCQs), short answers, or essays. These exams are designed to assess the ability to understand, synthesize, and express ideas in a logical and structured way;

• On the other hand,**oral exams** are frequently used in foreign language learning and to evaluate project presentations. They emphasize the ability to express oneself clearly and fluently, while also testing responsiveness and the ability to answer questions instantly. • Regarding **practical exams**, they are essential in technical, artistic, or physical fields. These exams assess specific practical skills, such as conducting laboratory experiments, performing music with an instrument, or demonstrating performance in a sport.

1.2.3 Historical Origins of Exams

Exams trace their origins back to ancient civilizations. In imperial China, as early as the Han Dynasty (206 BC – 220 AD), imperial examinations were a central tool for selecting civil servants, based on mastery of Confucian classics. This meritocratic system had a lasting impact on Chinese social and educational organization. In the West, the emergence of medieval universities introduced more structured forms of assessment, often in the form of oral debates and disputations. These practices aimed to evaluate students' understanding, memory, and ability to argue.[1]

1.3 Online Exams

1.3.1 Definition

An online exam is a type of virtual exam undertaken on a computer device (desktop, laptop) using the internet. One can give and take an online examination from any location remotely. Most online exams are invigilated either through human proctors or through various types of proctoring software or using a mix of the two. In an online examination, candidates are expected to answer the questions on their systems. The test window closes once the exam is over. An online exam does not have to be in an MCQ format – it can be written in any desired form.[2]

1.3.2 History of Online Exams

Online exams trace their origins back to educational platforms in the late 1990s and early 2000s, when systems like Moodle and Blackboard began allowing digital assessments in closed environments. However, it wasn't until the rise of e-learning in the 2010s — and more specifically, during the COVID-19 pandemic — that online exams became established as a real and necessary alternative for all types of educational and corporate institutions. Today, with the development of technologies such as artificial intelligence, remote proctoring, and biometrics, online exams have reached a level of maturity and reliability that makes them comparable to — and in some contexts even superior to — traditional assessments.[3]

1.4 Online Formative/Summative Evaluation System

1.4.1 Summative Assessment

The goal of summative assessment is to evaluate student learning at the end of an instructional unit by comparing it against some standard or benchmark. Summative assessments are often high stakes, which means that they have a high point value. Examples of summative assessments include: midterm exam,a final project,a paper, a senior recital... Information from summative assessments can be used formatively when students or faculty use it to guide their efforts and activities in subsequent courses.[4]

1.4.2 Formative Assessment

are in-process evaluations of student learning that are typically administered multiple times during a unit, course, or academic program. The general purpose of formative assessment is to give educators in-process feedback about what students are learning or not learning so that instructional approaches, teaching materials, and academic support can be modified accordingly. Formative assessments are usually not scored or graded, 10 and they may take a variety of forms, from more formal quizzes and assignments to informal questioning techniques and in-class discussions with students. [5]

1.4.3 Comparison Between Formative and Summative Assessment

Table 1.1: Comparison Between Formative and Summative Assessment

Aspect	Formative Assessment	Summative Assessment
Timing	During the training/learning process.	At the end of a module or a course of study.
Objective	To monitor progress and correct difficulties.	To attest and certify learning achievements.
Format	Quizzes, exercises, informal evaluations.	Final exams, standardized tests, graded projects.
Recipient	Primarily the learner.	Teachers, organizations, recruiters, HR.
Results	Immediate feedback, non-official.	Grades or official certificates.
Main Purpose	To improve ongoing learning.	To validate and officially recognize the level achieved.

1.4.4 Examples of Online Formative and Summative Assessment Systems

Moodle: Moodle is an open-source Learning Management System (LMS) created in 2002 in Australia by Martin Dougiamas. It allows for the design of online courses, quizzes, exams, and pedagogical activities. It is utilized for both formative evaluations (quizzes, self-assessments) and summative evaluations (final exams).[6]

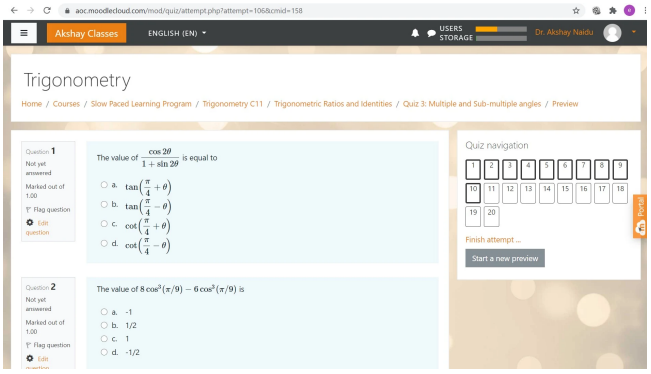


Figure 1.1: Trigonometry online test on moodle .

ExamOnline:ExamOnline is an education technology startup headquartered in India, founded in 2009. It offers a comprehensive solution for remote exam administration, with a strong emphasis on maintaining exam integrity and security. The startup's advanced AI-powered proctoring technology, secure browser capabilities, and customizable settings establish a seamless and user-friendly experience, setting a new standard for online assessments. Its features include live video monitoring, facial recognition, AI-based behavior analysis to prevent cheating, and ensuring exam fairness. The platform also provides services such as Remote Proctored Examination, Online Examination Software, Online Registration Software, and Processing as a Service. It also offers end-to-end online examination/assessment services, leveraging over 12 years of industry experience. [7]

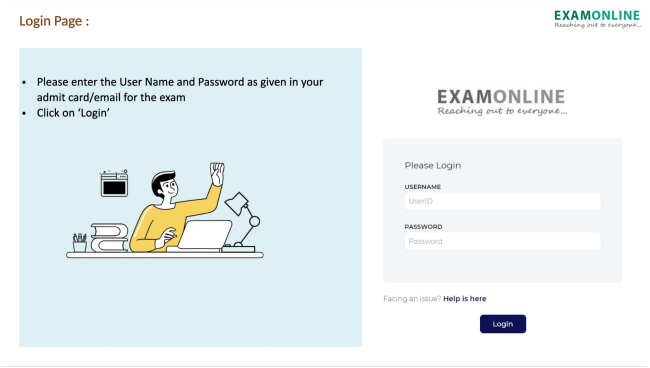


Figure 1.2: Login page of the platform .

Google Forms: Google Forms is a free survey and form-building tool launched in 2016 as a standalone application (originally integrated with Google Sheets) that is part of the Google Docs Editors suite. It allows users to create and distribute online forms, quizzes, and surveys, and collect responses in real time. While easy to use and accessible, Google Forms has limitations in customization, data handling, and advanced form functionality.[8]

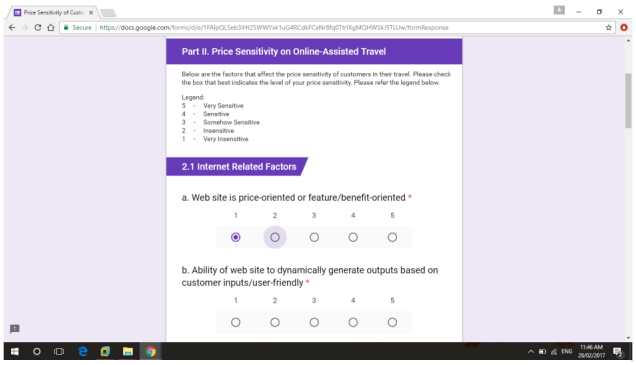


Figure 1.3: An Online Exam on GoogleForms .

EvalBox:Evalbox was founded in 2011 by Nadir Boussoukaia in France [9], It's a free online test maker service, built to create and manage online exams (or paper exams). It is an easy to use online 24/7 testing service, trusted by thousands of worldwide professionals, which embeds advanced automatic grading tools. You just have to simply create quizzes using the random quiz maker (that will randomly pick up questions from your question banks), or manually selecting the ones that fits the most with your needs, and schedule your exams. Its unique exam monitoring screen will help you to monitor / proctor online exams in real time. Thanks to advanced fraud detection and cheat prevention algorithms, you will be warned whenever a student is behaving strangely...[10]

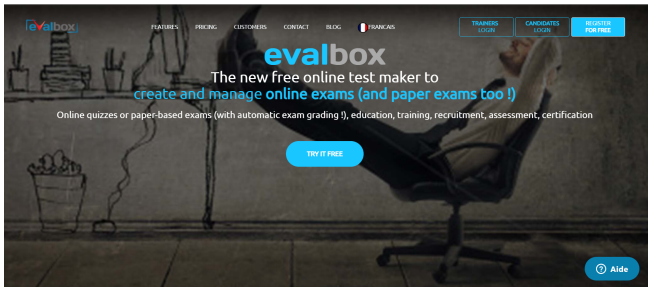


Figure 1.4: Main page of the platform.

ProctorU: ProctorU, established in 2008 in Hoover, Ala., was founded by Jarrod Morgan to proctor the online exams of Alabama-based Andrew Jackson University using basic webcams and screen-sharing technology. Andrew Jackson University, Troy University, California Southern University and Western Governors University were among the first institutions to employ the Hoover-based firm. ProctorU provides a full suite of online proctoring and identity management solutions for education, professional development and certification organizations. With patented, 24/7 on-demand live proctoring and a fully-automated platform, both backed by artificial intelligence, ProctorU offers a powerful, convenient and cost-effective alternative to traditional test centers. ProctorU increases access to online learning while ensuring exam integrity and accountability for any test-taker with internet access using only a computer and a webcam.[11]

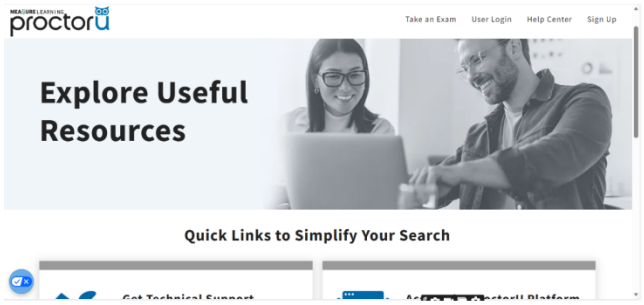


Figure 1.5: ProctorU landing page for remote exam supervision.

TestWe: TestWe is a French startup founded in 2014 in Paris. It is Europe's E-exam leader! It is an all-in-one e-Test solution, with an online exam builder for administrators and an offline software for applicants. TestWe offers candidates a secure, offline, and comfortable software that allows them to focus on their argumentation without feeling rushed. Most crucially, exam takers can take their exams in an exam room or remotely using their own device (BYOD) (live proctoring).[12]

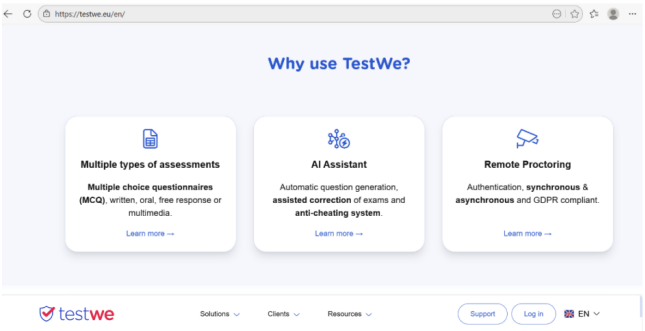


Figure 1.6: Functional overview of the TestWe online assessment solution.

1.5 Advantages and Challenges of Online Examinations

1.5.1 Advantages

- **Automated Grading:** Certain question types, such as multiple-choice or true/false, can be automatically graded by the system, reducing the evaluator's time and effort.
- **Large-Scale Group Management:** Platforms allow for the simultaneous management of a high volume of participants, which would be logistically challenging in a traditional setting.
- **Diversity of Question Types:** Online exams support various formats, including Multiple Choice Questions (MCQs), true/false, short answers, open-ended questions, and even interactive or multimedia exercises.
- **Integration of Technology and Pedagogy:** Online examinations combine technological tools with pedagogical strategies to enhance the efficiency and quality of the assessment process.

1.5.2 Challenges

- **Risk of Cheating:** The potential for academic dishonesty in the absence of adequate proctoring or supervision.
- **Technical Issues:** Dependence on stable internet connectivity and the risk of software malfunctions or system crashes.
- **Digital Divide:** Inequalities in the use of and access to digital technologies, referring to the disparity between individuals with internet access and the necessary digital literacy and those who lack such access or skills.

1.6 Conclusion

In this chapter, we have presented the essential concepts related to traditional and online examinations, as well as the two primary forms of evaluation: formative and summative. We have also demonstrated that online examination systems can be applied within both academic and professional fields. This theoretical framework constitutes an essential foundation for the design and development of our online examination system, which will be presented in detail in the following chapters.

Chapter 2

System Analysis And Conception

2.1 Introduction

After having presented the general context and the evolution of platforms for remote exams in Chapter 1, this chapter will be focused on the analysis and the specification of the proposed system. The goal of this chapter is to determine and specify all the different types of needs (functional, non functional) of the platform so that we can obtain an optimal, secure and user friendly solution. This chapter also intends to model the system by specifying how users interact with the platform and by defining the structure of the data used. We will reach this goal through a detailed needs analysis, then by identifying the principal actors who will interact with the system. Next, we will describe the functionalities of the system through use case diagrams. We will finally establish the data model using class diagrams. These diagrams are representative of the logical structure of the platform. This chapter represents a fundamental phase in our approach to development because it presents a clear and structured view of what we want to develop before we do it.

2.2 Needs Analysis

The needs assessment phase represents an initial crucial step in developing the Online Exam platform. This first stage identifies and defines each requirement which this website will need to fulfill so as to be satisfactory for use by those who have access to it. Requirements are grouped together into two main categories: requirements related to functionality and requirements unrelated to functionality (non-functionality).

2.2.1 Functional Needs

Functional requirements describe the main services and operations that the system must provide. The online examination platform should ensure the following functionalities:

- The system must allow users to log in securely.
- The system must differentiate between different types of users (Student, Professor, Administrator).
- The Professor must be able to create, edit, and delete exams.
- The Professor must be able to add different types of questions (multiple choice, open-ended, etc.).
- The student must be able to access and pass exams within a specified time.
- The system must include a timer to control the duration of the exam.
- The system must automatically correct objective questions such as multiple-choice questions.
- The system must allow students to view their results after submission.
- The administrator must be able to manage users and oversee the system.

2.2.2 Non-functional Needs

Non-functional requirements define the quality attributes and constraints of the system. The platform should meet the following criteria:

- **Security**: The system must ensure the protection of user data and prevent unauthorized access, as well as minimize the risk of cheating during examinations.
- **Performance**: The system should provide fast response times and be capable of handling multiple users simultaneously without degradation in performance.
- **Usability**: The user interface must be simple, intuitive, and easy to use for all categories of users.
- **Availability**: The platform should be accessible at all times with minimal downtime.
- **Scalability**: The system should be able to accommodate an increasing number of users and exams without affecting its performance.
- **Reliability**: The system must operate correctly and consistently, especially during exam sessions, without errors or failures.

This analysis provides a clear understanding of what the system is expected to achieve and serves as a foundation for the design phase presented in the following sections.

2.3 Actors Identification

In order to better understand the interactions within the online examination platform, it is essential to identify the different actors involved in the system. An actor represents any user or external entity that interacts with the system to achieve specific goals.

The main actors of the system are as follows:

- **Student**: The student is the primary user of the platform. They are responsible for accessing the system, consulting available exams, passing examinations within a defined time, and viewing their results after submission.
- **Professor**: The Professor is responsible for managing the examination content. They can create, edit, and delete exams, as well as add and manage different types of questions. They may also review students results.
- **Administrator**: The administrator oversees the overall functioning of the platform. They manage user accounts, assign roles, and ensure the proper operation and maintenance of the system.

These actors interact with the system through various functionalities, which will be illustrated in the use case diagrams presented in the following section.

2.4 Use Case Diagram

A Use Case Diagram is a type of Unified Modeling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals. It provides a high-level view of the system's functionality by illustrating the various ways users can interact with it.[13]

General Use Case Diagram

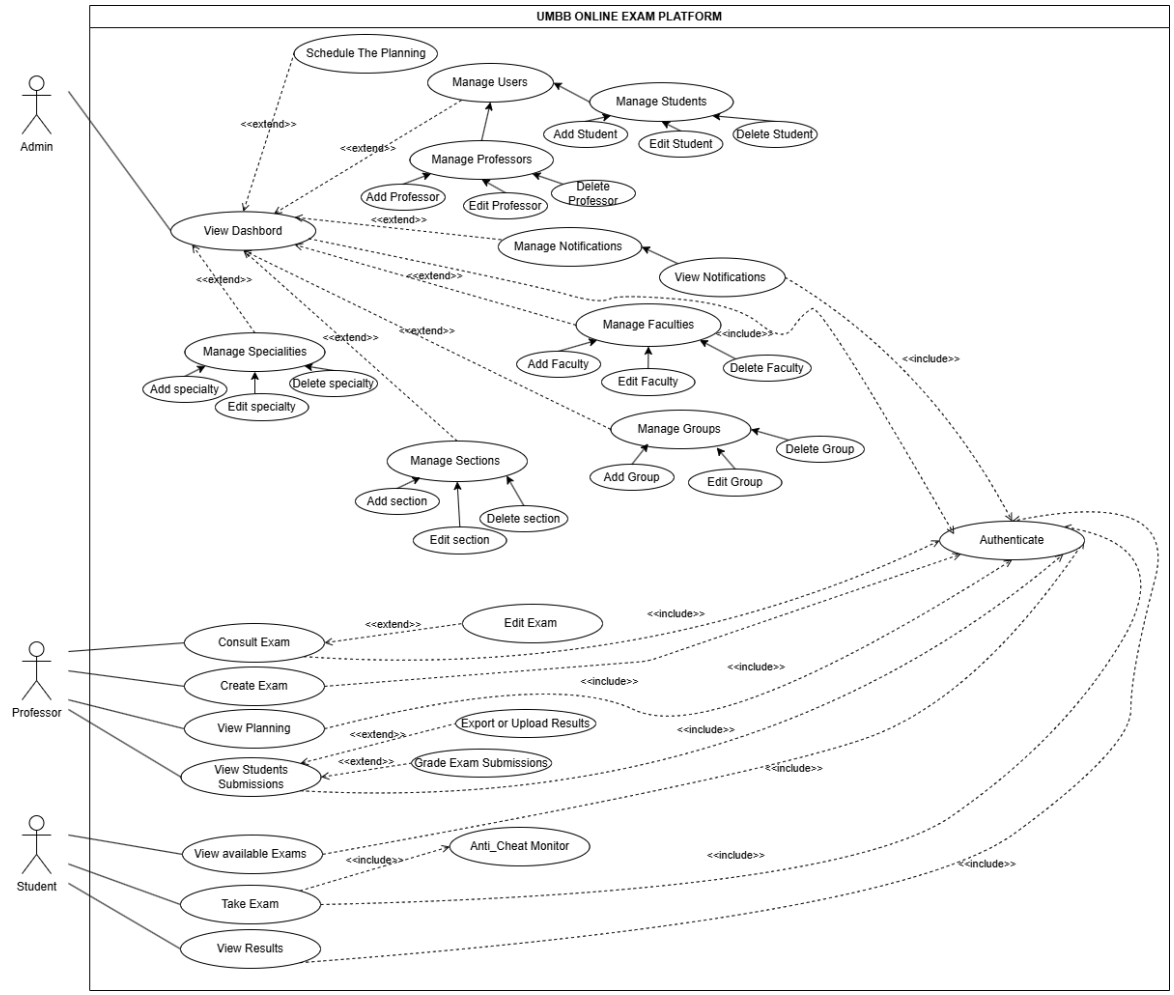


Figure 2.1: General Use Diagram .

2.5 Sequence Diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that maps out how users and systems interact over time. It's commonly used in interaction design and human-computer interaction studies to show what happens, when, and between which parts of a system.[14]

2.5.1 Sequence Diagram: Authenticate

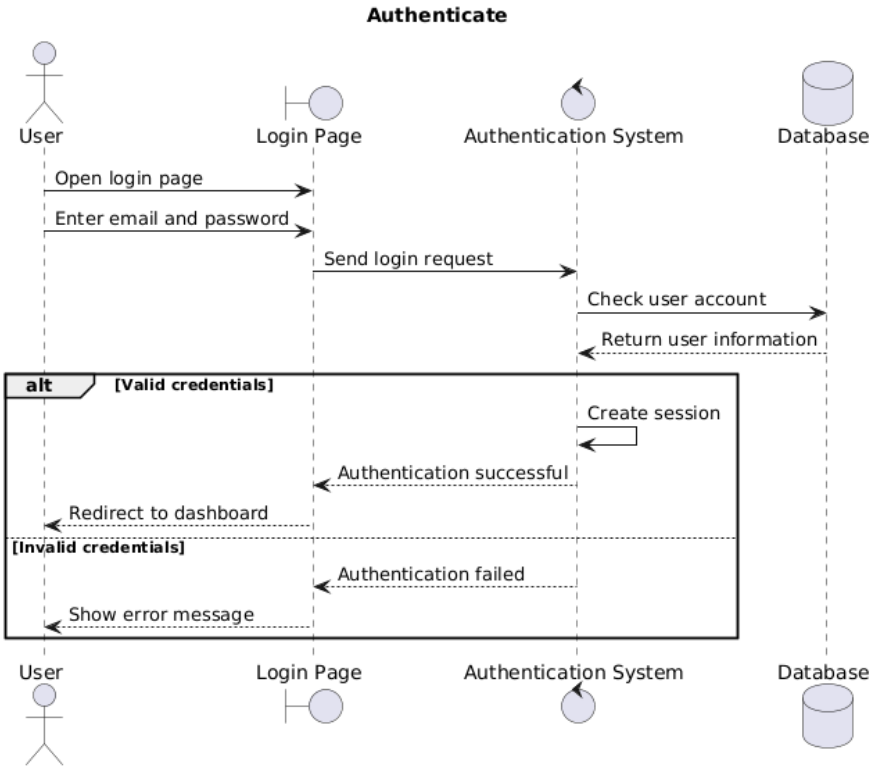


Figure 2.2: Sequence Diagram For Authentication

Use Case: Authenticate	
Principal Actor	Student, Professor, Administrator
Objective	Allow users to log into the platform and access their role-specific dashboard.
Precondition	The user has a valid registered account.
Nominal Sequence	1. User opens the login page. 2. User enters email and password. 3. System verifies the credentials. 4. System creates an authenticated session. 5. User is redirected to the corresponding dashboard.
Alternative	1. Invalid credentials → error message displayed. 2. Missing fields → system requests completion.
Postcondition	User is authenticated and session is stored.

Table 2.1: Use Case Description: Authenticate

2.5.2 Sequence Diagram: Manage Notifications

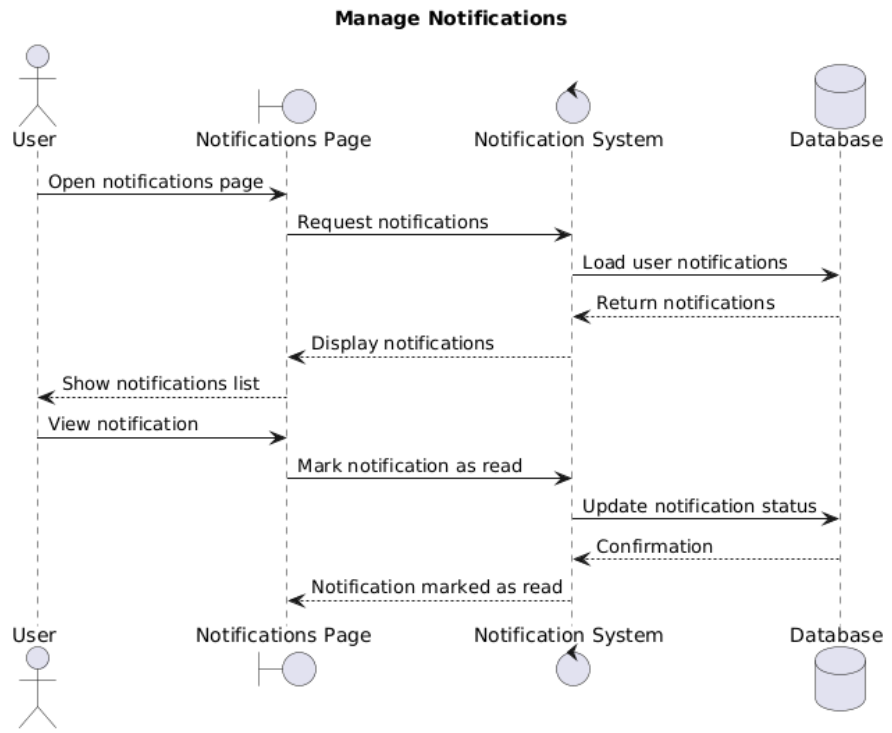


Figure 2.3: Sequence Diagram for Notifications Management

Use Case: Manage Notifications	
Principal Actor	Administrator, Professor, Student
Objective	Allow users to view notifications related to planning, exams, and results.
Precondition	User is authenticated.
Nominal Sequence	1. User opens the notifications page. 2. System retrieves notifications for the user. 3. System displays title, message, type, and date. 4. System marks viewed notifications as read.
Alternative	No notifications found → empty state message displayed.
Postcondition	User has viewed notifications.

Table 2.2: Use Case Description: Manage Notifications

2.5.3 Sequence Diagram: Administrator

2.5.3.1 Sequence Diagram: Manage Users

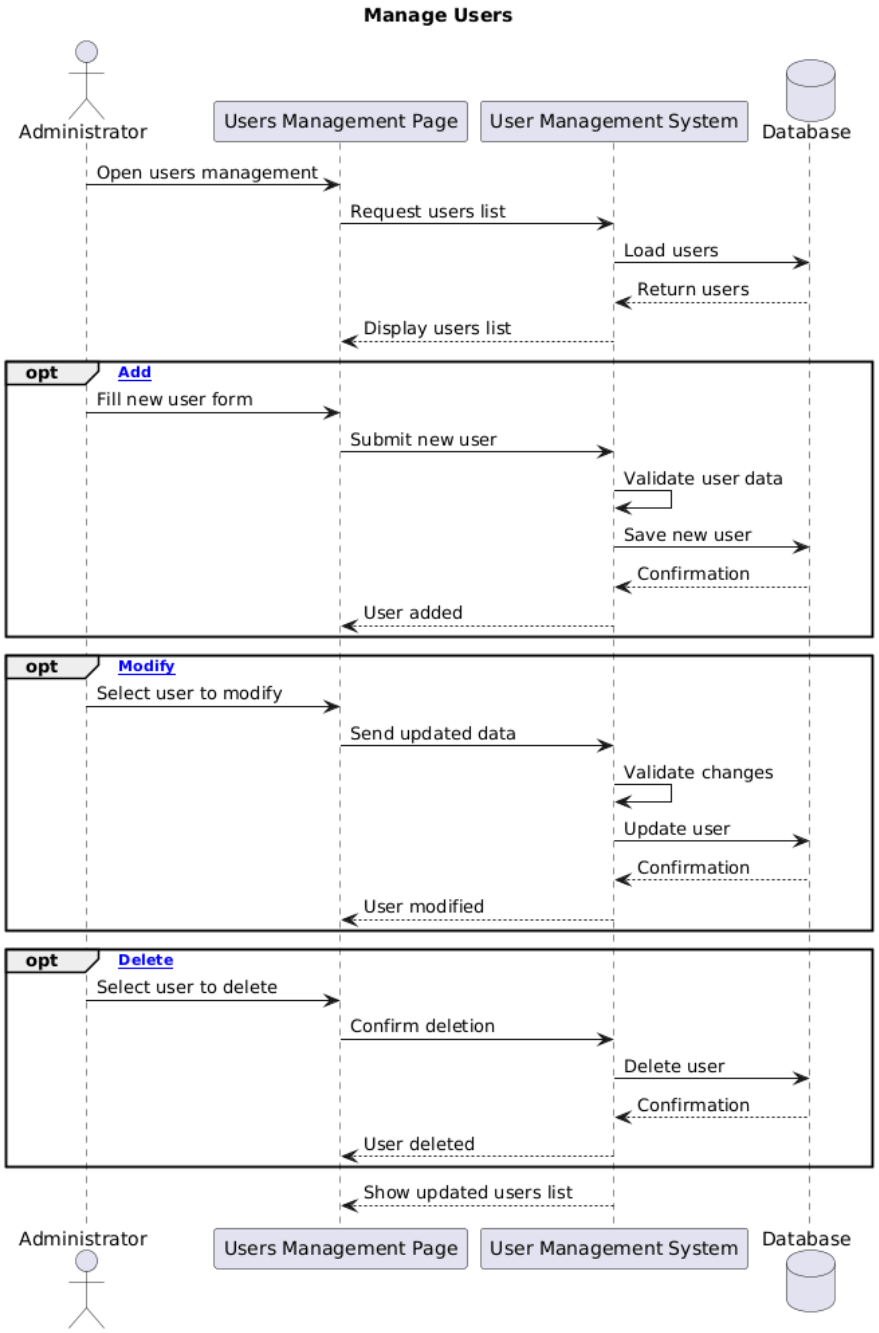


Figure 2.4: Sequence Diagram for Users Management

Use Case: Manage Users	
Principal Actor	Administrator
Objective	Allow the administrator to manage student and professor accounts.
Precondition	Administrator is authenticated.
Nominal Sequence	1. Administrator opens the user management page. 2. System displays the list of users. 3. Administrator adds, modifies, or deletes a user. 4. System validates the entered data. 5. System saves the changes in the database.
Alternative	1.Administrator searches or filters users before acting. 2.Required data missing → validation error displayed. 3. Email already exists → operation rejected.
Postcondition	User records are updated.

Table 2.3: Use Case Description: Manage Users

2.5.3.2 Sequence Diagram: Schedule Planning

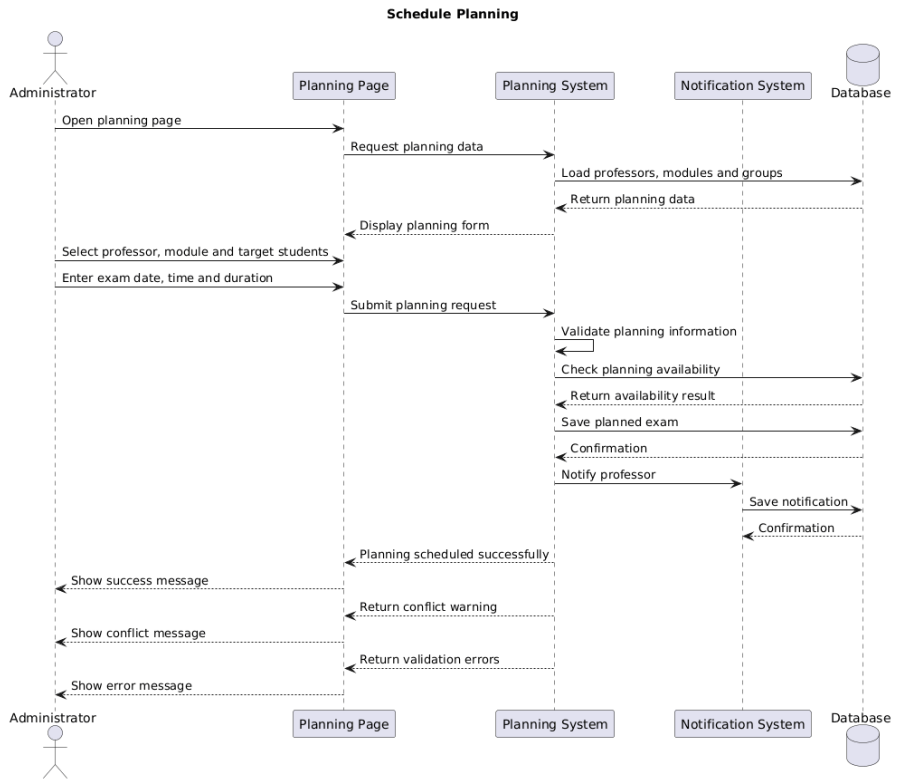


Figure 2.5: Sequence Diagram for planning scheduling

Use Case: Schedule Planning	
Principal Actor	Administrator
Objective	Allow the administrator to plan exams before professors create content.
Precondition	Administrator is authenticated. Professors and modules exist.
Nominal Sequence	1. Administrator opens the planning page. 2. Administrator selects professor, module, target group, date, time, and duration. 3. System validates the planning data. 4. System creates the planned exam. 5. System notifies the concerned professor.
Alternative	1. Administrator reschedules an existing planned exam. 2. Date or time in the past → planning rejected. 3. Conflict with another exam → administrator warned.
Postcondition	Exam planning saved with status `planned`. Professor notified.

Table 2.4: Use Case Description: Schedule Planning

2.5.4 Sequence Diagram: Professor

2.5.4.1 Sequence Diagram: Create Exam

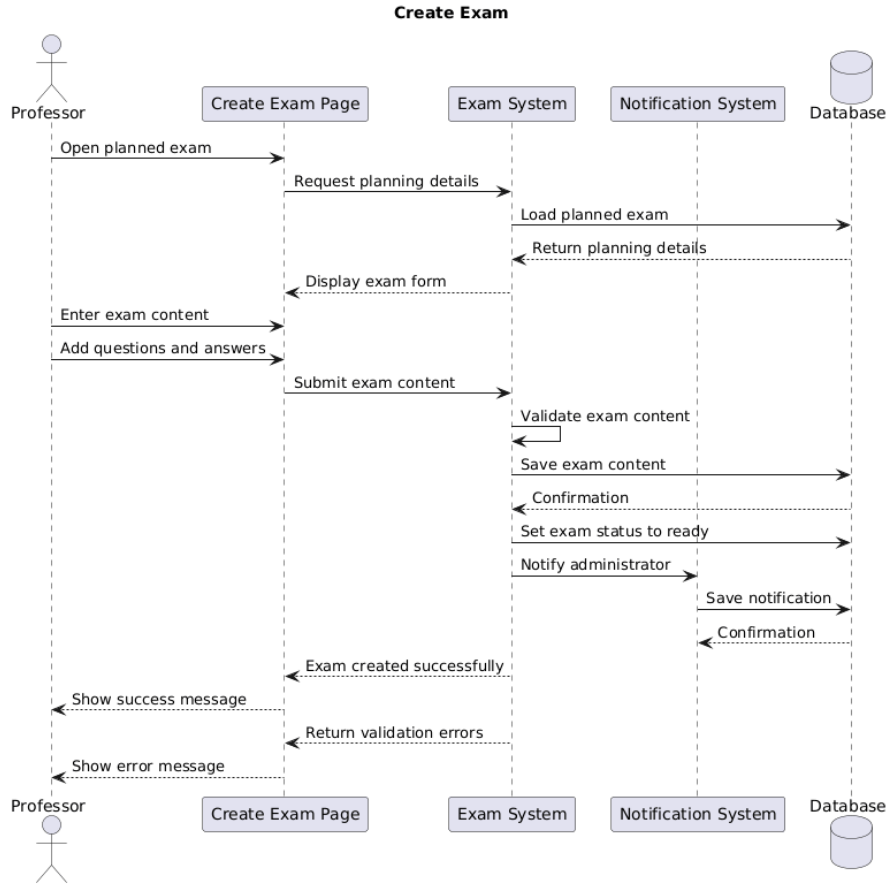


Figure 2.6: Sequence Diagram for Exam Creation

Use Case: Create Exam	
Principal Actor	Professor
Objective	Allow the professor to create exam content for a planned exam.
Precondition	Professor is authenticated. A planned exam is assigned.
Nominal Sequence	1. Professor selects a planned exam. 2. System displays planning info and exam form. 3. Professor enters title, description, type, and questions. 4. Professor adds answer options for MCQ questions if needed. 5. System saves content and sets status to `ready`. 6. System notifies the administrator.
Alternative	1. Professor saves incomplete content as draft. 2. No question added → submission prevented. 3. Invalid question data → validation errors displayed.
Postcondition	Exam content saved and ready for administrator approval.

Table 2.5: Use Case Description: Create Exam

2.5.4.2 Sequence Diagram: Consult Exam

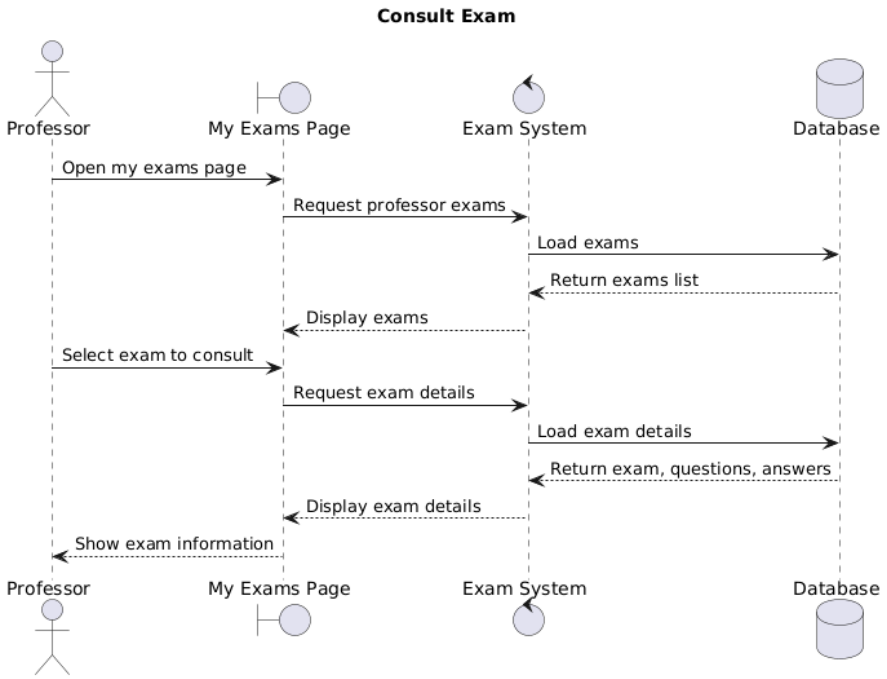


Figure 2.7: Sequence Diagram for Exams Consultation

2.5.4.3 Sequence Diagram: Modify Exam

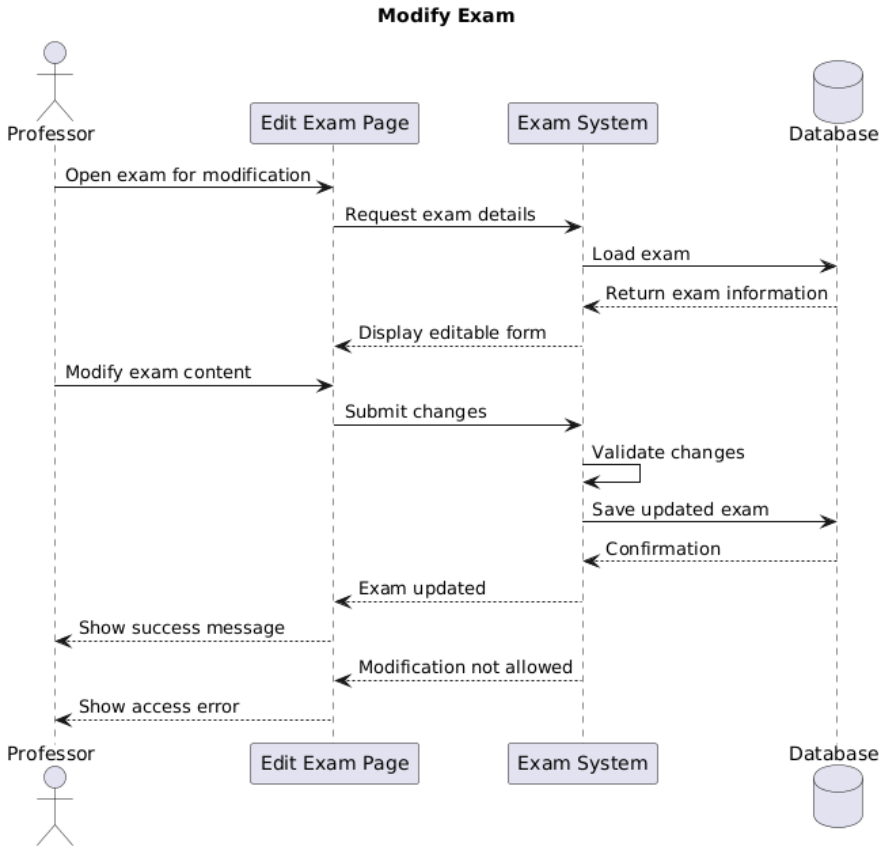


Figure 2.8: Sequence Diagram for Exam Editing

Use Case: Consult / Modify Exam	
Principal Actor	Professor
Objective	Allow the professor to consult or modify exam content before publication.
Precondition	Professor is authenticated and owns the exam.
Nominal Sequence	1. Professor opens the exam list. 2. Professor selects an exam to consult or modify. 3. System displays exam details and questions. 4. Professor modifies allowed fields. 5. System saves the updates.
Alternative	1.Professor only consults without modifying. 2. Exam is published or closed → modification blocked.
Postcondition	Exam content consulted or updated.

Table 2.6: Use Case Description: Consult / Modify Exam

2.5.5 Sequence Diagram: Student

Sequence Diagram: Take Exam

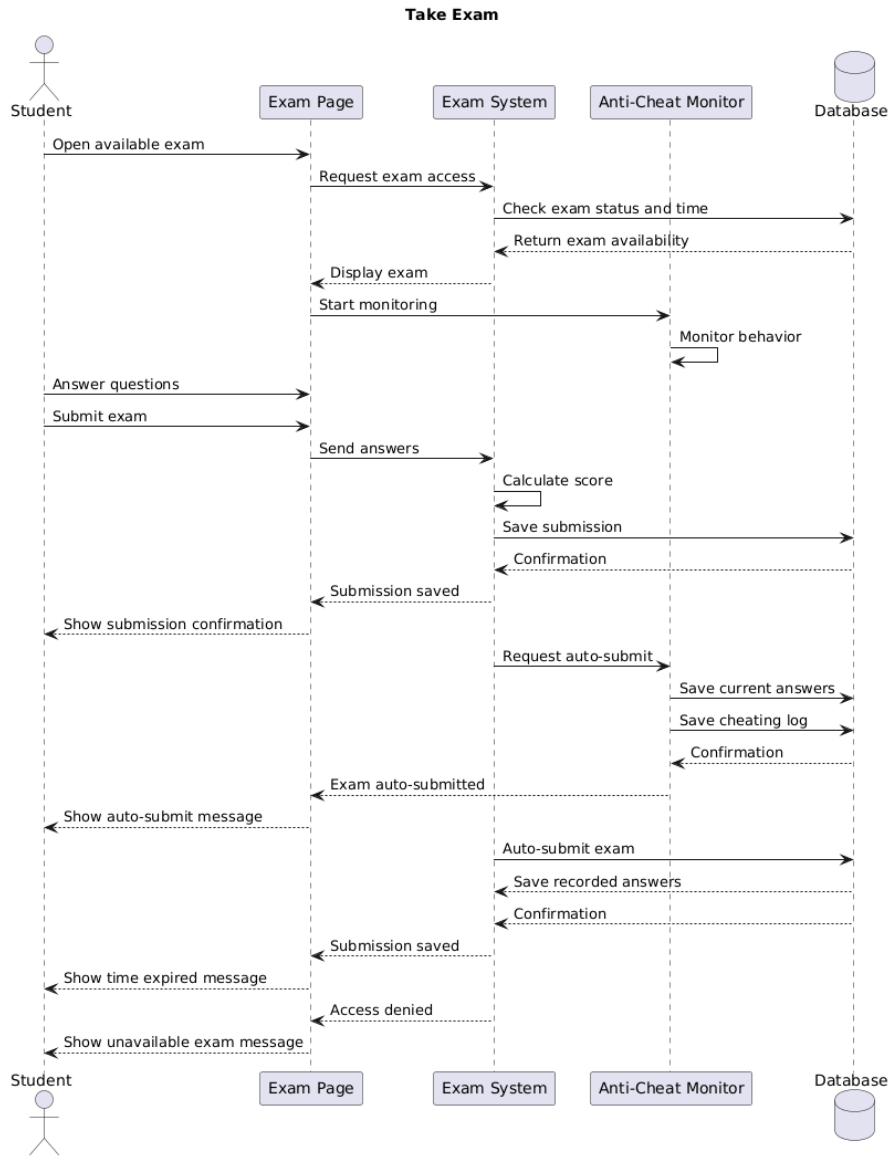


Figure 2.9: Sequence Diagram for taking an Exam

Use Case: Take Exam	
Principal Actor	Student
Objective	Allow the student to answer and submit an exam during the allowed time window.
Precondition	Student is authenticated. Exam is published and currently open.
Nominal Sequence	1. Student selects an available exam. 2. System verifies access time and eligibility. 3. System loads the exam submission. 4. Anti-cheat monitor starts monitoring the session. 5. Student answers questions and submits the exam. 6. System records answers and calculates score.
Alternative	1. Time expires → system auto-submits recorded answers. 2. Exam not open → access denied. 3. Max warnings reached → exam auto-submitted.
Postcondition	Submission saved with status `submitted`.

Table 2.7: Use Case Description: Take Exam

2.5.6 Use Cases Description Tables

Use Case: View Dashboard	
Principal Actor	Administrator
Objective	Display an overview of platform activity, users, exams, and notifications.
Precondition	Administrator is authenticated.
Nominal Sequence	1. Administrator opens the dashboard page. 2. System retrieves statistics from the database. 3. System displays users count, exams count, planning queue, and notifications.
Alternative	Database unavailable → dashboard data cannot be loaded.
Postcondition	Administrator views the current platform overview.

Table 2.8: Use Case Description: View Dashboard

Use Case: Manage Academic Structure	
Principal Actor	Administrator
Objective	Manage faculties, specialties, sections, and groups.
Precondition	Administrator is authenticated.
Nominal Sequence	1. Administrator selects an academic entity to manage. 2. System displays existing records. 3. Administrator adds, modifies, or deletes a record. 4. System validates the hierarchy links. 5. System saves the changes in the database.
Alternative	1. Administrator cancels the operation before saving. 2. Parent entity missing → creation prevented. 3. Entity has dependent records → deletion prevented.
Postcondition	Academic structure is updated.

Table 2.9: Use Case Description: Manage Academic Structure

Use Case: View Planning	
Principal Actor	Professor
Objective	Allow the professor to view exams planned by the administrator.
Precondition	Professor is authenticated and has planned exams assigned.
Nominal Sequence	1. Professor opens the planning page. 2. System retrieves planned exams assigned to the professor. 3. System displays module, date, time, duration, and status.
Alternative	1. No planning exists → empty list displayed. 2. Database error → planning cannot be displayed.
Postcondition	Professor identifies exams requiring content creation.

Table 2.10: Use Case Description: View Planning

Use Case: View Student Submissions	
Principal Actor	Professor
Objective	Allow the professor to view submitted exams and student scores.
Precondition	Professor is authenticated and has exams with submissions.
Nominal Sequence	1. Professor opens exam submissions. 2. System retrieves submitted exam records. 3. System displays student identity, score, warnings, and date.
Alternative	1. No submissions exist → empty state displayed. 2. Professor does not own the exam → access denied.
Postcondition	Professor views student submissions.

Table 2.11: Use Case Description: View Student Submissions

Use Case: Grade Exam Submissions	
Principal Actor	Professor
Objective	Allow the professor to grade open-ended answers.
Precondition	Professor is authenticated. Submission contains open-ended answers.
Nominal Sequence	1. Professor opens a submitted exam. 2. System displays student answers and question points. 3. Professor assigns scores to open-ended answers. 4. System recalculates the total score. 5. System saves the grading results.
Alternative	1. Only MCQ answers present → professor views automatic grading only. 2. Score exceeds question points → value rejected.
Postcondition	Submission score updated.

Table 2.12: Use Case Description: Grade Exam Submissions

Use Case: Export / Upload Results	
Principal Actor	Professor
Objective	Allow the professor to export or upload corrected student results.
Precondition	Professor is authenticated and owns the exam.
Nominal Sequence	1. Professor opens the submissions page. 2. Professor chooses to export results. 3. System generates a results file. 4. Professor may upload a corrected results file. 5. System validates and updates student scores.
Alternative	1. Professor exports results without uploading corrections. 2. Invalid file format → upload rejected.
Postcondition	Results exported or updated.

Table 2.13: Use Case Description: Export / Upload Results

Use Case: View Available Exams	
Principal Actor	Student
Objective	Allow the student to view published exams assigned to their group, section, cycle, or year.
Precondition	Student is authenticated.
Nominal Sequence	1. Student opens the exams page. 2. System checks the student's academic information. 3. System retrieves matching published exams. 4. System displays available and upcoming exams.
Alternative	1. No exam available → empty state displayed. 2. Missing academic assignment → no matching exams displayed.
Postcondition	Student views eligible exams.

Table 2.14: Use Case Description: View Available Exams

Use Case: View Results	
Principal Actor	Student
Objective	Allow the student to consult their submitted exam results.
Precondition	Student is authenticated and has submitted at least one exam.
Nominal Sequence	1. Student opens the results page. 2. System retrieves submitted exams and scores. 3. System displays exam title, module, date, and score.
Alternative	1. No submitted exams → empty result list displayed. 2. Results unavailable → message displayed.
Postcondition	Student views available results.

Table 2.15: Use Case Description: View Results

2.6 Data Modeling (Class Diagram)

A class diagram is a type of UML (Unified Modeling Language) diagram that visually represents the structure of a system by showing its classes, attributes, methods, and relationships. It serves as a blueprint for how different parts of an object-oriented system interact and organize data. Class diagrams are fundamental in system design because they bridge the gap between real-world concepts and code implementation. This helps developers, analysts, and stakeholders understand how the system's components are connected and how they collaborate to perform various functions.[15]

2.6.1 Data Dictionary

Table 2.16: Data Dictionary — UniExam UMBB

Data Name	Format	Length	Class
id	Numeric	11	User
first_name	Alphabetic	100	/
last_name	Alphabetic	100	/
email	Alphanumeric	180	/
password	Alphanumeric	255	/
role	Alphanumeric	20	/
cycle	Alphanumeric	20	/
id	Numeric	11	Faculty
name	Alphabetic	200	/
id	Numeric	11	Department
name	Alphabetic	200	/
id	Numeric	11	Specialty
name	Alphabetic	200	/
id	Numeric	11	Section
name	Alphanumeric	100	/
id	Numeric	11	Group
name	Alphanumeric	100	/
id_module	Numeric	11	Module
name	Alphabetic	200	/
id_planning	Numeric	11	Planning
exam_date	Date	–	/
exam_time	Time	–	/
duration_minutes	Numeric	4	/
status	Alphanumeric	20	/
id_exam	Numeric	11	Exam
title	Alphanumeric	250	/
description	Alphanumeric	1000	/
entry_window_minutes	Numeric	4	/
exam_type	Alphanumeric	20	/
id_question	Numeric	11	Question
question_text	Alphanumeric	1000	/
points	Numeric	6,2	/
order_num	Numeric	11	/
question_type	Alphanumeric	20	/
id_answerOption	Numeric	11	AnswerOption

Data Name	Format	Length	Class
option_text	Alphanumeric	500	/
is_correct	Boolean	1	/
id_examSubmission	Numeric	11	ExamSubmission
submitted_at	DateTime	–	/
submitted_date	Date	–	/
total_score	Numeric	6,2	/
warnings_count	Numeric	4	/
is_finished	Boolean	1	/
status	Alphanumeric	30	/
id	Numeric	11	StudentAnswer
answer_text	Alphanumeric	2000	/
score	Numeric	6,2	/
id_notification	Numeric	11	Notification
title	Alphanumeric	250	/
message	Alphanumeric	1000	/
is_read	Boolean	1	/
type	Alphanumeric	30	/
hasWarnings	Numeric	4	AntiCheatMonitor
isFaceRecognized	Boolean	1	/
isTabSwitchDetected	Boolean	1	/
isFullscreenActive	Boolean	1	/
isCopyPasteDetected	Boolean	1	/
isRightClickDisabled	Boolean	1	/
isMicrophoneActive	Boolean	1	/
isMultipleFaceDetected	Boolean	1	/
isSuspiciousBehavior	Boolean	1	/

2.6.2 Business Rules

Table 2.17: Business Rules — UniExam UMBB

Rule Code	Business Rule Description	Concerned Classes
BS-01	Each admin, professor, and student inherits common user info: name, email, password, role, and cycle.	`User, Admin, Professor, Student`
BS-02	A user account corresponds to one role: administrator, professor, or student.	`User, Admin, Professor, Student`
BS-03	A faculty contains one or more departments; each department belongs to one faculty.	`Faculty, Department`
BS-04	A department contains one or more specialties; each specialty belongs to one department.	`Department, Specialty`
BS-05	A specialty contains one or more sections; each section belongs to one specialty.	`Specialty, Section`
BS-06	A section contains one or more groups; each group belongs to one section.	`Section, Group`
BS-07	A group contains students; each student belongs to one group.	`Group, Student`
BS-08	A student has a cycle used to determine exam eligibility.	`Student, User, Exam`
BS-09	A professor creates, edits, deletes, and closes exams.	`Professor, Exam`
BS-10	A professor may view exam planning, view submissions, export results, and grade submissions.	`Professor, Planning, ExamSubmission`
BS-11	An administrator manages users, planning, faculties, and notifications.	`Admin, Planning, Faculty, Notification`
BS-12	A planning record defines exam date, time, duration, and status.	`Planning`
BS-13	Each planning record has exactly one exam, and each exam belongs to one planning record.	`Planning, Exam`

Rule Code	Business Rule Description	Concerned Classes
BS-14	Each exam covers one module; a module may be covered by one or more exams.	`Module, Exam`
BS-15	Planning status may be: `planned`, `draft`, `pending`, `early_approved`, `published`, or `closed`.	`Planning`
BS-16	An exam may be of type `online`, `onsite`, or `hybrid`.	`Exam`
BS-17	An exam must contain one or more questions.	`Exam, Question`
BS-18	A question belongs to one exam.	`Question, Exam`
BS-19	A question may be MCQ or open-ended.	`Question`
BS-20	A question may have zero or more answer options.	`Question, AnswerOption`
BS-21	Each answer option belongs to one question and is marked correct or incorrect.	`AnswerOption, Question`
BS-22	An exam may have zero or more submissions.	`Exam, ExamSubmission`
BS-23	Each submission records one or more student answers.	`ExamSubmission, StudentAnswer`
BS-24	A student answer is linked to one question, and a question may have zero or more student answers.	`StudentAnswer, Question`
BS-25	A student answer may reference zero or more answer options, and each answer option relates to one student answer.	`StudentAnswer, AnswerOption`
BS-26	A student may view available exams, take exams, and view their results.	`Student, Exam, ExamSubmission`
BS-27	A user may receive zero or more notifications.	`User, Notification`
BS-28	An exam may generate zero or more notifications.	`Exam, Notification`
BS-29	Notification types: `exam_planned`, `exam_created`, `exam_ready`, `exam_approved`, `exam_published`, `general`.	`Notification`
BS-30	Each exam submission may have zero or more anti-cheat monitoring records.	`ExamSubmission, AntiCheatMonitor`
BS-31	The anti-cheat monitor tracks detection flags including face recognition, tab switching, fullscreen state, copy-paste, right-click, microphone activity, multiple-face detection, and suspicious behavior, and can flag violations.	`AntiCheatMonitor`

2.6.3 Class Diagram:

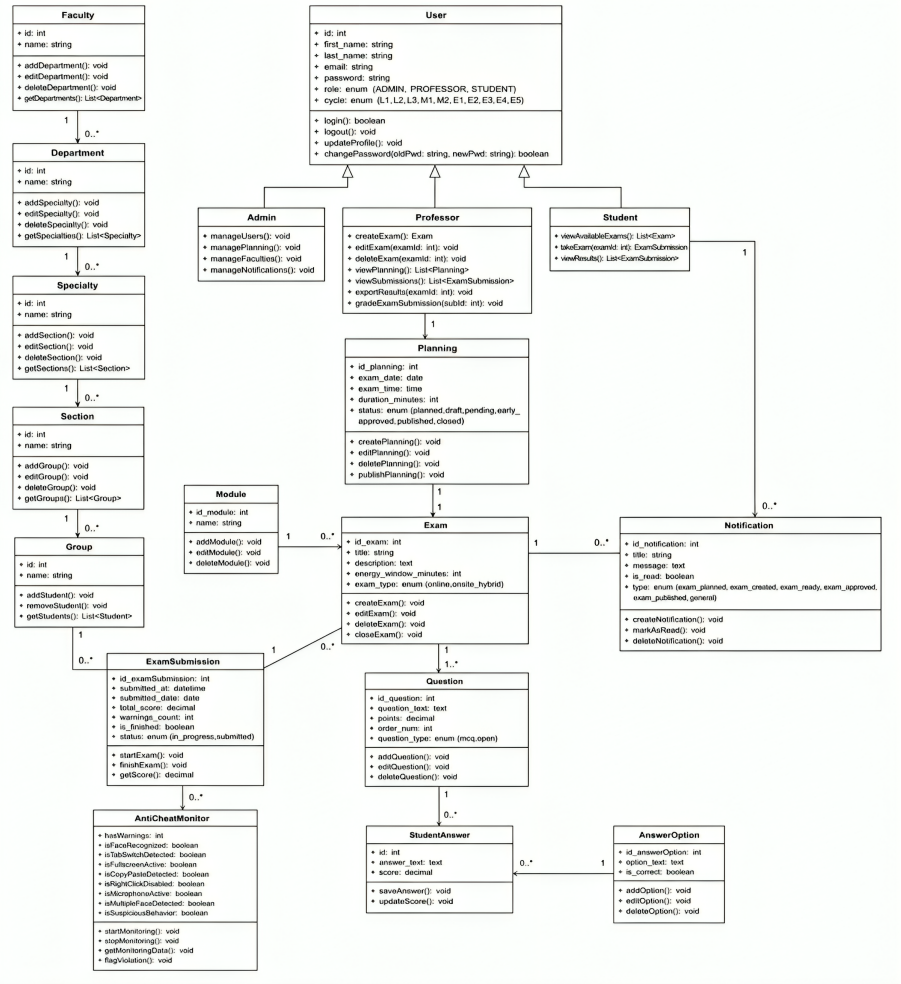


Figure 2.10: Class Diagram

2.6.4 Database Relational Model

- **User** (id_user, first_name, last_name, email, password, role, cycle)
- **Admin** (#id_user)
- **Professor** (#id_user)
- **Student** (#id_user, #id_group)
- **Faculty** (id_faculty, name)
- **Department** (id_department, name, #id_faculty)
- **Specialty** (id_specialty, name, #id_department)
- **Section** (id_section, name, #id_specialty)
- **Group** (id_group, name, #id_section)
- **Module** (id_module, name, #id_specialty)
- **ProfessorModule** (id, #id_professor, #id_module)
- **Planning** (id_planning, exam_date, exam_time, duration_minutes, status, #id_admin, #id_module)

- **Exam** (id_exam, title, description, energy_window_minutes, exam_type, #id_planning, #id_professor, #id_module)
- **ExamSection** (#id_exam, #id_section)
- **ExamGroup** (#id_exam, #id_group)
- **Question** (id_question, question_text, points, order_num, question_type, #id_exam)
- **AnswerOption** (id_answerOption, option_text, is_correct, #id_question)
- **ExamSubmission** (id_examSub, submitted_at, submitted_date, total_score, warnings_count, is_finished, status, #id_exam, #id_student)
- **StudentAnswer** (id, answer_text, score, #id_examSub, #id_question, #id_answerOption)
- **AntiCheatMonitor** (id_monitor, hasWarnings, isFaceRecognized, isTabSwitchDetected, isFullscreenActive, isCopyPasteDetected, isRightClickDisabled, isMicrophoneActive, isSuspiciousBehavior, #id_examSub)
- **Notification** (id_notification, title, message, is_read, type, #sender_id, #receiver_id, #related_exam_id)

2.7 Rules of Passage: Class Diagram → Relational Model

Introduction

The rules of passage define the systematic process of transforming a UML class diagram into a relational database model. Each element of the class diagram — classes, attributes, associations, and inheritance relationships — is converted into tables and foreign keys following a set of well-defined rules. The following section presents the six rules applied to derive the relational model of our system.

Rule 1 — Each Class Becomes a Table

General Rule: Every class in the class diagram is transformed into a relational table. The attributes of the class become the columns of that table, and the identifier attribute (underlined in the diagram) becomes the Primary Key (PK).

Application: The following classes are each transformed into an independent table:

Class	Resulting Table (Primary Key)
User	User (id_user PK, first_name, last_name, email, password, role, cycle)
Faculty	Faculty (id_faculty PK, name)
Department	Department (id_department PK, name)
Specialty	Specialty (id_specialty PK, name)
Section	Section (id_section PK, name)
Group	Group (id_group PK, name)
Module	Module (id_module PK, name)
Planning	Planning (id_planning PK, exam_date, exam_time, duration_minutes, status)
Exam	Exam (id_exam PK, title, description, energy_window_minutes, exam_type)
Question	Question (id_question PK, question_text, points, order_num, question_type)
AnswerOption	AnswerOption (id_answerOption PK, option_text, is_correct)
ExamSubmission	ExamSubmission (id_examSub PK, submitted_at, submitted_date, total_score, warnings_count, is_finished, status)
StudentAnswer	StudentAnswer (id PK, answer_text, score)
Notification	Notification (id_notification PK, title, message, is_read, type)
AntiCheatMonitor	AntiCheatMonitor (id_monitor PK, hasWarnings, isFaceRecognized, isTabSwitchDetected, isFullscreenActive, isCopyPasteDetected, isRightClickDisabled, isMicrophoneActive, isSuspiciousBehavior)

Table 2.18: Rule 1 — Class to Table Transformation

Rule 2 — Inheritance → Foreign Key in Child Table

General Rule: When a class inherits from a parent class, the primary key of the parent becomes a foreign key (FK) in the child table. The child table does not duplicate the parent's attributes; it only references it via the foreign key.

Application: The three classes Admin, Professor, and Student all inherit from User. Each child table contains the foreign key #id_user referencing the User table.

Child Class	Resulting Table
Admin	Admin (#id_user → User)
Professor	Professor (#id_user → User)
Student	Student (#id_user → User, #id_group → Group)

Table 2.19: Rule 2 — Inheritance Transformation

Rule 3 — One-to-Many Association (1:N) → Foreign Key

General Rule: When a 1:N association exists between two classes, the primary key of the class on the "1" side migrates into the table of the class on the "N" side as a foreign key. This foreign key represents the link between the two tables.

Application: The following 1:N associations were identified and transformed:

Association (1 → N)	Foreign Key Added
Faculty (1) → Department (N)	Department gets #id_faculty
Department (1) → Specialty (N)	Specialty gets #id_department
Specialty (1) → Section (N)	Section gets #id_specialty
Section (1) → Group (N)	Group gets #id_section
Specialty (1) → Module (N)	Module gets #id_specialty
Group (1) → Student (N)	Student gets #id_group
Planning (1) → Exam (N)	Exam gets #id_planning, #id_professor, #id_module
Admin (1) → Planning (N)	Planning gets #id_admin, #id_module
Exam (1) → Question (N)	Question gets #id_exam
Question (1) → AnswerOption (N)	AnswerOption gets #id_question
Exam (1) → ExamSubmission (N)	ExamSubmission gets #id_exam, #id_student
ExamSubmission (1) → StudentAnswer (N)	StudentAnswer gets #id_examSub, #id_question, #id_answerOption
ExamSubmission (1) → AntiCheatMonitor (1)	AntiCheatMonitor gets #id_examSub

Table 2.20: Rule 3 — One-to-Many Transformation

Rule 4 — Many-to-Many Association (N:M) → Intermediate Table

General Rule: When a N:M association exists between two classes, it cannot be directly represented by a simple foreign key. A new intermediate table is created, containing the primary keys of both classes as foreign keys. This table may also include additional attributes describing the relationship.

Application: Two N:M associations were identified in the class diagram:

Association (N ↔ M)	Intermediate Table Created
Exam (N) ↔ Section (M)	ExamSection (#id_exam → Exam, #id_section → Section)
Exam (N) ↔ Group (M)	ExamGroup (#id_exam → Exam, #id_group → Group)

Table 2.21: Rule 4 — Many-to-Many Transformation

Rule 5 — Multiple Associations to the Same Table → Distinct Foreign Keys

General Rule: When a class holds two or more associations pointing to the same table, each association is represented by a distinct foreign key with a different name to avoid ambiguity.

Application: The Notification class is associated with User twice (once as the sender, once as the receiver), and also references the Exam table:

Association	Foreign Key
Notification → User (as sender)	#sender_id → User
Notification → User (as receiver)	#receiver_id → User
Notification → Exam (related exam)	#related_exam_id → Exam

Table 2.22: Rule 5 — Multiple Associations Transformation

Summary — Final Relational Model

The application of the five rules above produces the following 21 tables:

#	Table	Primary Key	Foreign Keys
1	User	id_user	—
2	Admin	id_user	#id_user → User
3	Professor	id_user	#id_user → User
4	Student	id_user	#id_user → User, #id_group → Group
5	Faculty	id_faculty	—
6	Department	id_department	#id_faculty → Faculty
7	Specialty	id_specialty	#id_department → Department
8	Section	id_section	#id_specialty → Specialty
9	Group	id_group	#id_section → Section
10	Module	id_module	#id_specialty → Specialty
11	Planning	id_planning	#id_admin → Admin, #id_module → Module
12	Exam	id_exam	#id_planning → Planning, #id_professor → Professor, #id_module → Module
13	ExamSection	—	#id_exam → Exam, #id_section → Section
14	ExamGroup	—	#id_exam → Exam, #id_group → Group
15	Question	id_question	#id_exam → Exam
16	AnswerOption	id_answerOption	#id_question → Question
17	ExamSubmission	id_examSub	#id_exam → Exam, #id_student → Student
18	StudentAnswer	id	#id_examSub → ExamSubmission, #id_question → Question, #id_answerOption → AnswerOption
19	AntiCheatMonitor	id_monitor	#id_examSub → ExamSubmission
20	Notification	id_notification	#sender_id → User, #receiver_id → User, #related_exam_id → Exam

Table 2.23: Summary — Final Relational Model (20 Tables)

2.8 Conclusion

This chapter established the complete architectural and functional blueprint for our system. By moving from a detailed requirements analysis to UML behavioral modeling (use cases and sequence diagrams) and structural design (class and relational models), we successfully translated user needs into concrete technical specifications. With the database logic and interaction flows fully defined, the system is now ready for the implementation and deployment phase detailed in the next chapter.

Chapter 3

Implementation

3.1 Introduction

Following the design and architectural modeling detailed in the previous chapter, this implementation phase focuses on the concrete development and deployment of the platform. This chapter describes the technical environment selected for the project, justifies the choice of the database management system (DBMS), details the implementation architecture, and showcases the main interfaces of the finalized application.

3.2 Software Environment

3.2.1 Development Tools

- **PlantUml**:PlantUML is a component that allows you to create various UML diagrams through simple textual descriptions. From sequence diagrams to deployment diagrams and beyond, PlantUML provides an easy way to create visual representations of complex systems. [16]

- **Drawio**:draw.io is a technology stack for building diagramming applications, and the world's most widely used browser-based end-user diagramming software. [17]

- **VS Code**: Visual Studio Code is a free, lightweight but powerful source code editor that runs on your desktop and on the web and is available for Windows, macOS, Linux, and Raspberry Pi OS. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other programming languages (such as C++, C#, Java, Python, PHP, and Go), runtimes (such as .NET and Unity), environments (such as Docker and Kubernetes), and clouds (such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform). [18]

- **WAMP**: WAMP (Windows, Apache, MySQL, and PHP) is a free software bundle that turns a Windows computer into a private, offline web server, allowing creators to build and test websites locally without internet lag or public risk. Serving as a safe "offline sandbox," it is an ideal tool for learning platforms like WordPress and testing code before spending money on hosting. Ultimately, it enables a professional workflow where a website can be fully perfected on a local machine (‘http://localhost‘) and then seamlessly migrated to a live, public server once finished. [19]

3.2.2 Development Languages

- **SQL**:Structured Query Language (SQL) is a database language designed for managing data held in a relational database management system. SQL was initially developed by IBM in the early 1970s (Date 1986). The initial version, called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's quasi-relational database management system, System R. Then in the late 1970s, Relational Software Inc., which is now Oracle Corporation, introduced the first commercially available implementation of SQL, Oracle V2 for VAX computers. Many of the currently available relational DBMSs, such as Oracle Database, Microsoft SQL Server, MySQL, IBM DB2, IBM Informix and Microsoft Access, use SQL.[20]

- **HTML** :HTML (Hypertext Markup Language) is a simple markup coding language used to layout content on a webpage. You use HTML programming language to tell web browsers how to display text, headings, images and other multimedia in the layout you want your users to see it in. HTML is platform-agnostic, meaning it can display web pages on any platform whether it's Windows, Mac Os, Linux or a mobile device operating system.[21]
- **CSS**: CSS, or Cascading Style Sheets, in a programming language that offers stylistic choices for web design. Deepen your understanding of Cascading Style Sheets (CSS), one of the key technologies in website creation.[22]
- **PHP** : PHP (recursive acronym for PHP: Hypertext Pre-processor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.[23]
- **JAVASCRIPT** :JavaScript is a browser-based programming language that brings interactivity to static web pages, applications and more. From animations to content updates, JavaScript is one of the three core web technologies, working alongside HTML and CSS to deliver dynamic web experiences . JavaScript is used for its ability to bring content to life, allowing developers to introduce almost endless dynamic options and objects to their websites.Add animations, introduce news tickers, display video, create pop-up chat windows – whatever 'live' element you see and interact with when online, it will most likely have been generated using JavaScript. [24]

3.3 Choice of DBMS (Database Management System)

phpMyAdmin is a free software tool used for managing MySQL and MariaDB databases through a web interface. It simplifies database administration with an easy-to-use graphical interface, making it accessible for users of all skill levels. The tool allows for various operations such as creating, modifying, and deleting databases and tables. It also supports executing SQL queries and managing user accounts, making it a versatile choice for web developers and database administrators.[25]

- **Database and Table Management:** It allows users to easily create, drop, alter, and browse databases, tables, fields, and indexes through intuitive visual menus.
- **SQL Query Execution:** It includes a built-in SQL editor where users can write, execute, and debug raw SQL queries, bookmarks, and complex join statements.
- **Data Import and Export:** It supports importing and exporting entire databases or individual tables in various formats—such as SQL (for migrations), CSV, XML, PDF, and JSON.
- **User Privileges Administration:** It provides robust security management tools to create user accounts, assign specific permissions (e.g., SELECT, INSERT, administrative rights), and manage passwords globally.
- **Visual Database Relations (Designer):** It features a visual "Designer" tool that can automatically generate interactive entity-relationship diagrams (ERD) based on foreign key mappings between tables.
- **Server Status Monitoring:** It offers real-time charts and metrics to monitor server health, track active connections, analyze query performance, and optimize database configuration settings.

3.4 Deployment Diagram

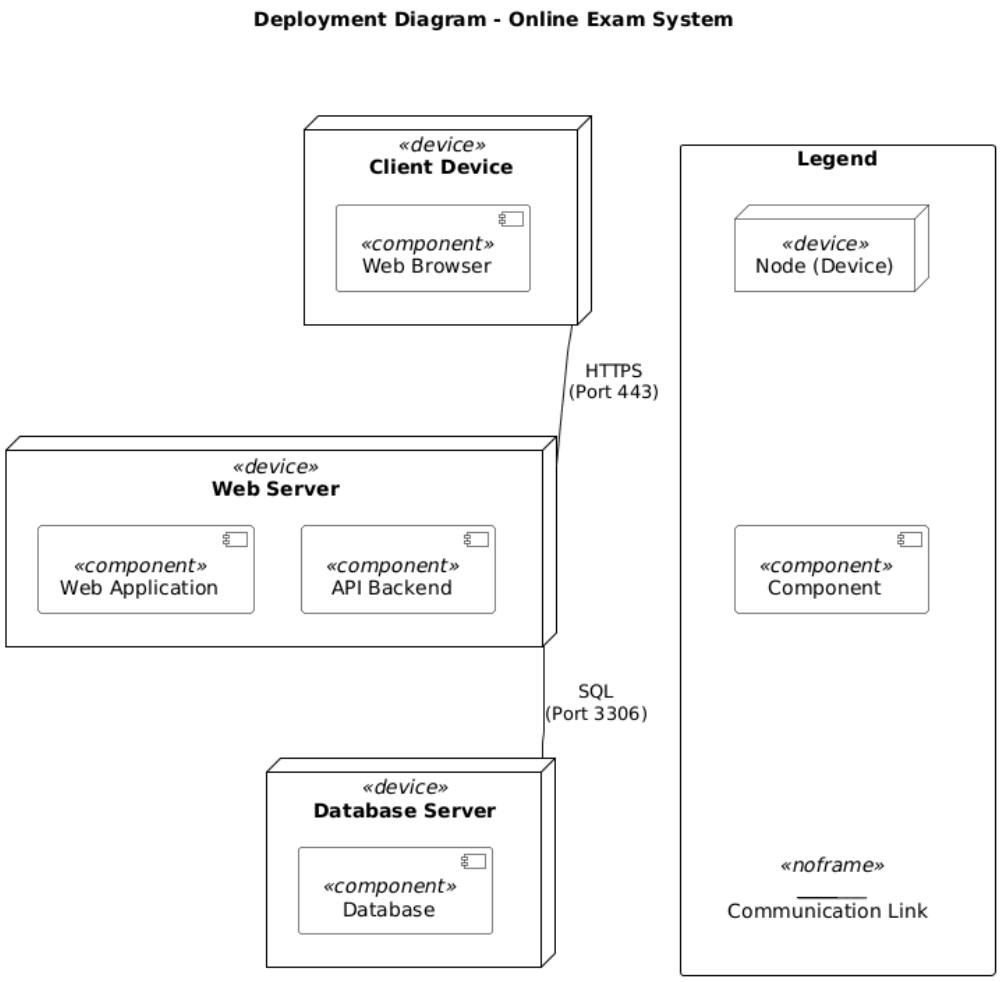


Figure 3.1: Deployment Diagram of the Online Examination System.

The deployment diagram illustrates the physical architecture of the Online Examination System and the interactions between its main hardware and software components.

The system follows a three-tier architecture composed of a client device, a web server, and a database server. Users, including students, teachers, and administrators, access the application through a web browser installed on the client device. Communication between the client and the web server is established through the HTTPS protocol (Port 443), ensuring secure data transmission.

The web server hosts the Online Examination System, which is developed using PHP, HTML, CSS, and JavaScript. This server is responsible for handling the application's business logic, including user management, exam management, question management, online examination sessions, automatic grading, and result generation.

The database server hosts a MySQL database that stores all application data, such as users, roles, exams, questions, options, answers, examination attempts, and results. Communication between the web server and the database server is performed through SQL requests using Port 3306.

The Internet/Network component represents the communication infrastructure that enables data exchange between all deployed nodes. This deployment architecture ensures scalability, maintainability, data integrity, and secure access to the Online Examination System.

3.5 Graphic Interfaces Presentation

3.5.1 Authentication Interface

The authentication interface allows students, professors, and administrators to log in using their credentials. The system redirects each user to their role-specific dashboard upon successful login.

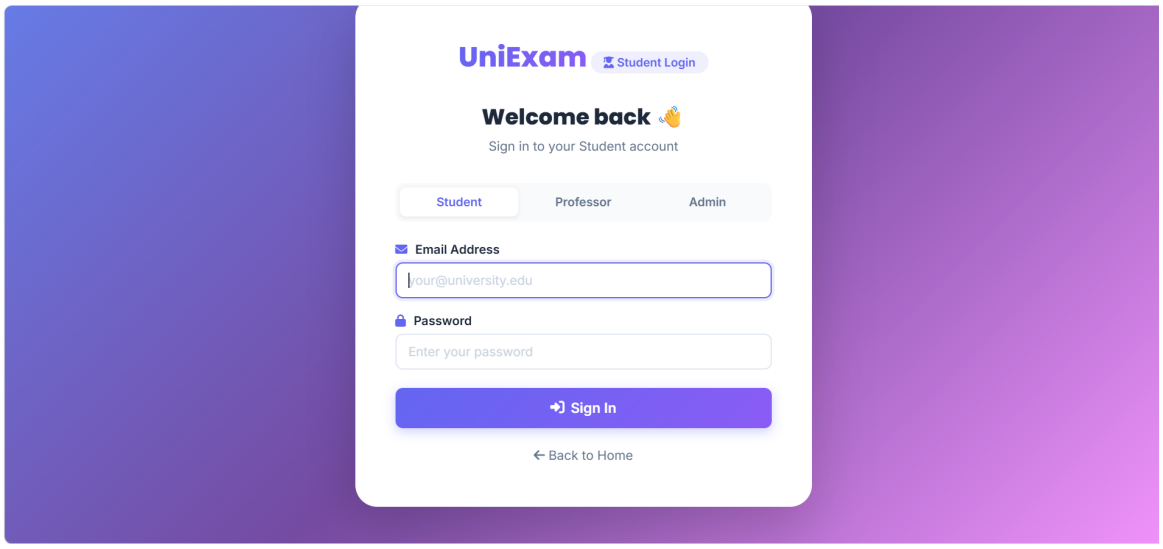


Figure 3.2: Authentication Interface.

3.5.2 Notifications Managing Interfaces

The notifications interface displays system alerts related to exam planning, submissions, and results for all user roles. Notifications are marked as read once viewed and updated accordingly in the database.

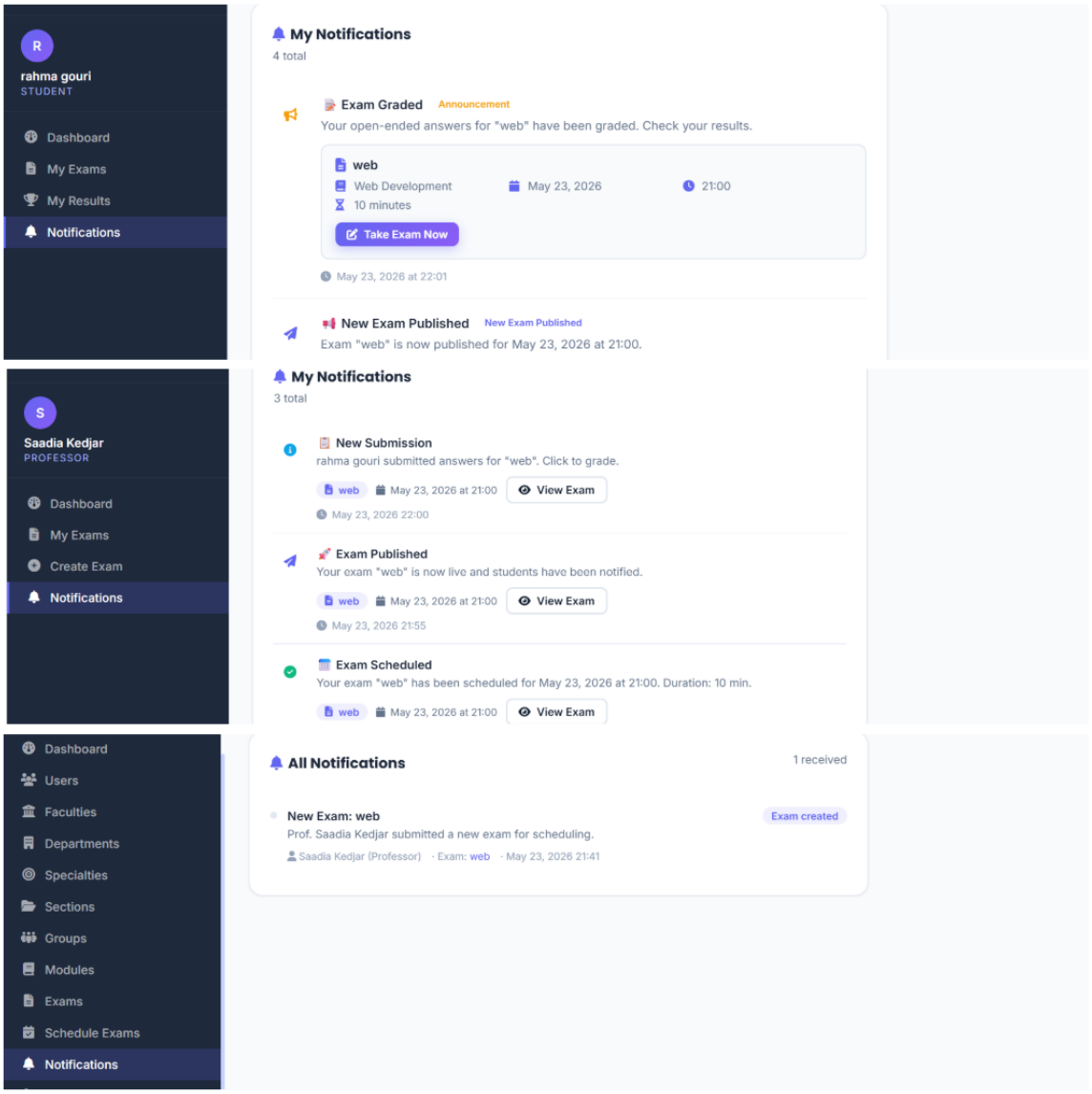


Figure 3.3: Notifications Managing Interfaces.

3.5.3 Admin Space

- **Exam planning Scheduling Interface**
  This interface enables the administrator to schedule exams by selecting the professor, module, target group, date, time, and duration. Planned exams are saved and a notification is automatically sent to the concerned professor.

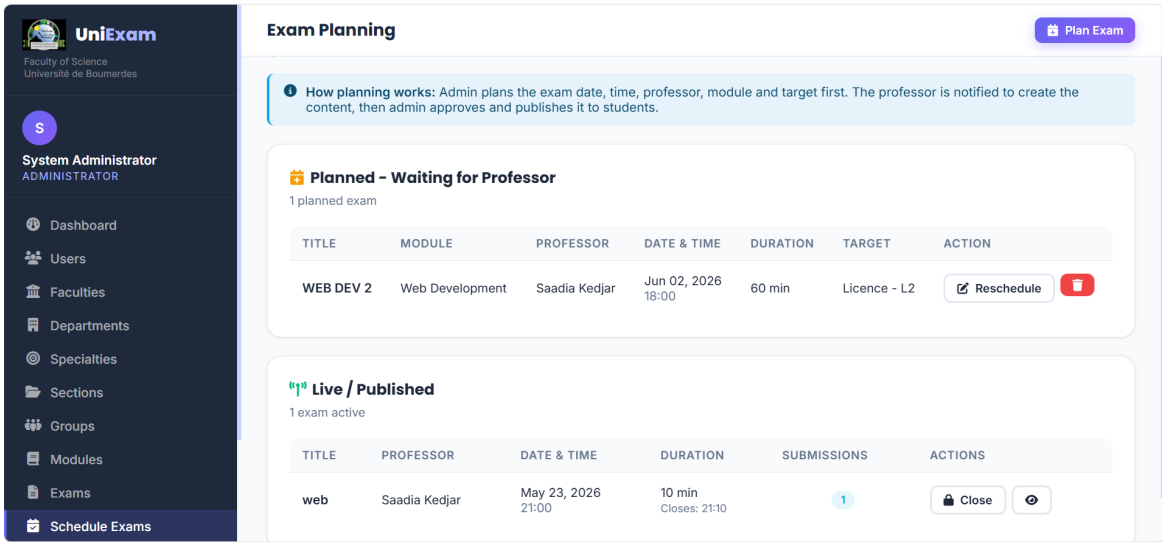


Figure 3.4: Exam planning Scheduling Interface.

- **Users Managing Interface**
  This interface allows the administrator to add, modify, and delete student and professor accounts. A search and filter functionality is provided to simplify navigation through the user list.

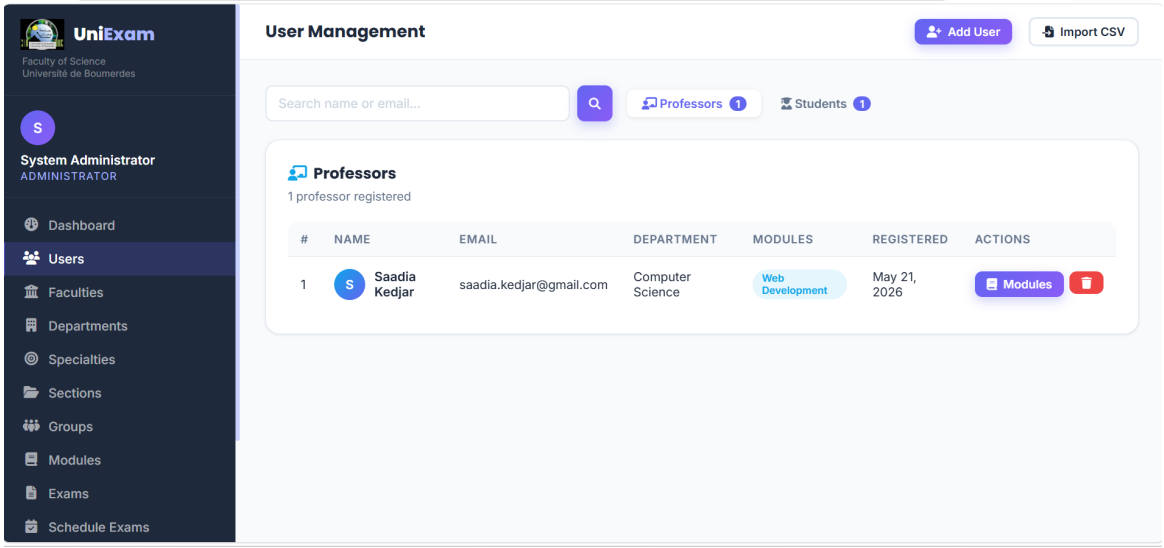


Figure 3.5: Users Managing Interface.

3.5.4 Professor Space

- **Exam Consultation Interface**
  Professors can view all their assigned exams along with key details such as module, date, duration, and current status. Action buttons allow direct access to exam creation, editing, and submission review.

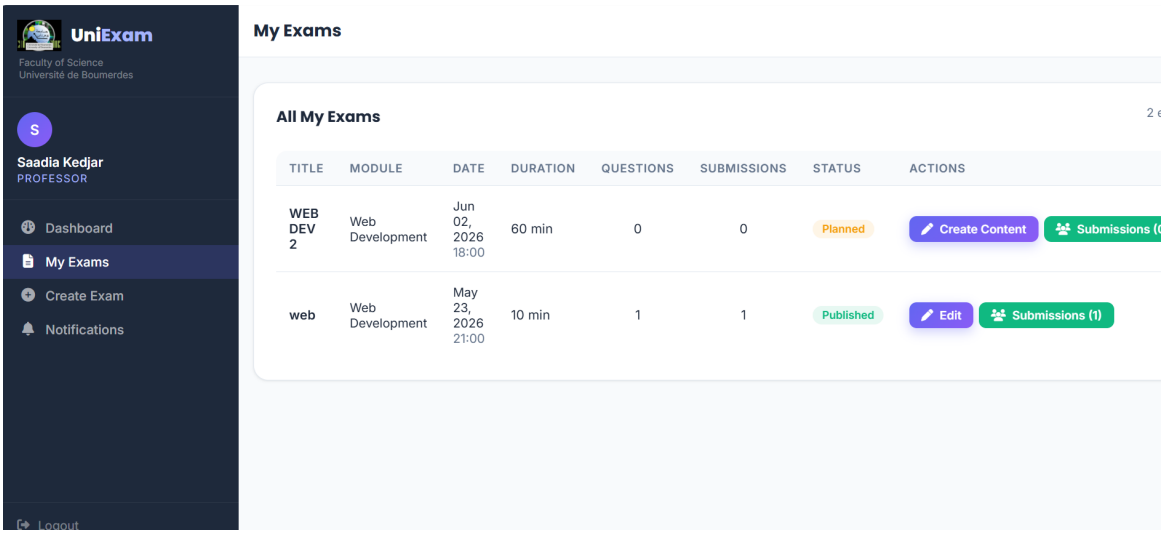


Figure 3.6: Exam Consultation Interface .

- **Exam Creation Interface**
  This interface allows the professor to fill in exam content for a planned exam, including title, description, question type, and individual questions. The exam status is updated to ready upon successful submission.

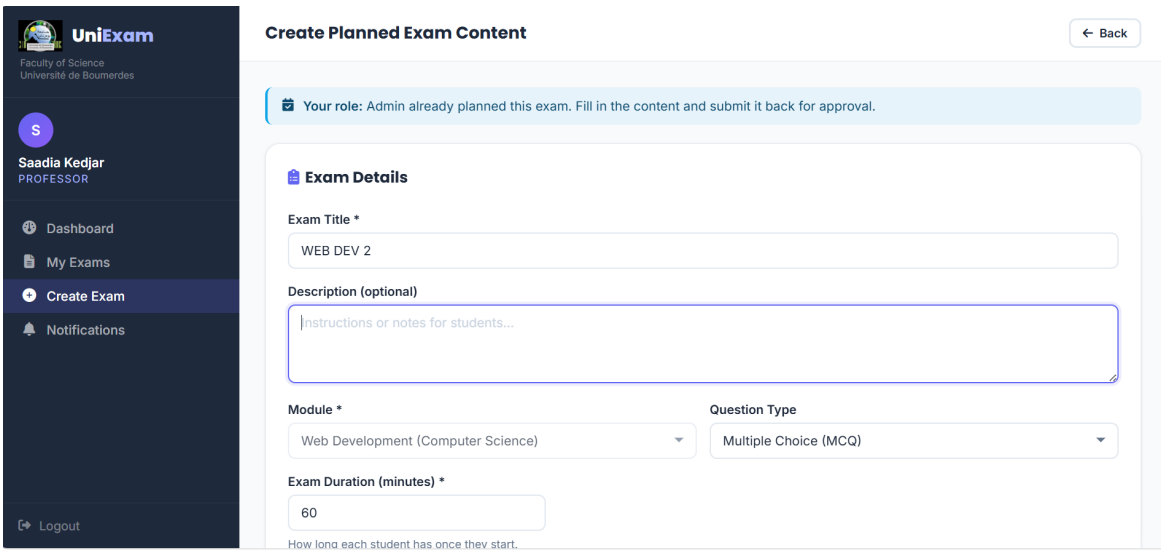


Figure 3.7: Exam Creation Interface .

- **Exam Editing Interface**
  Professors can modify exam details and questions through this interface as long as the exam has not been published or closed. Any edits made to a published exam automatically notify the admin and targeted students.

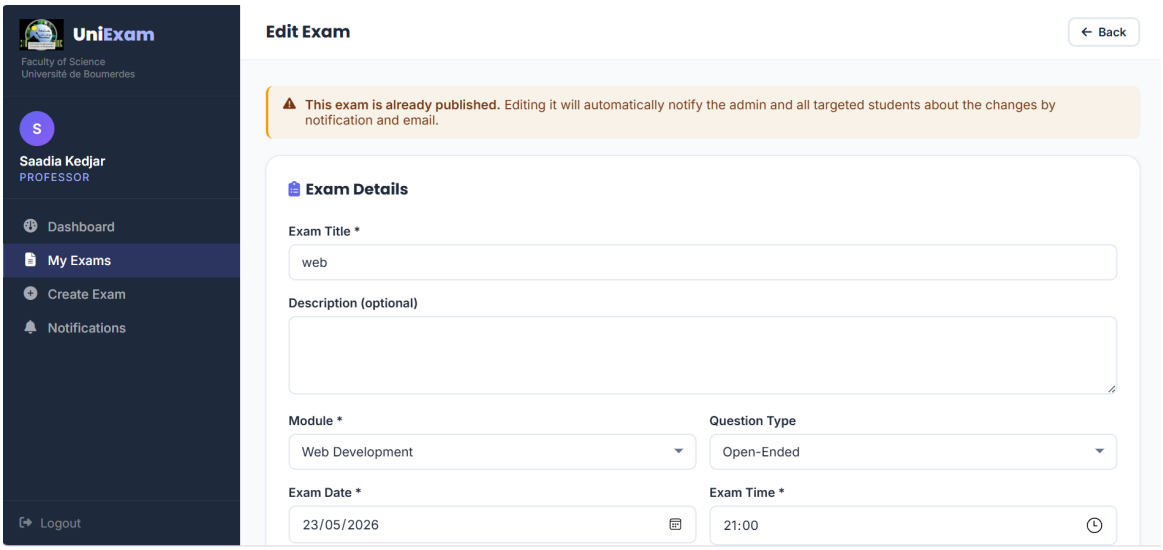


Figure 3.8: Exam Editing Interface.

3.5.5 Student Taking Exam Interface

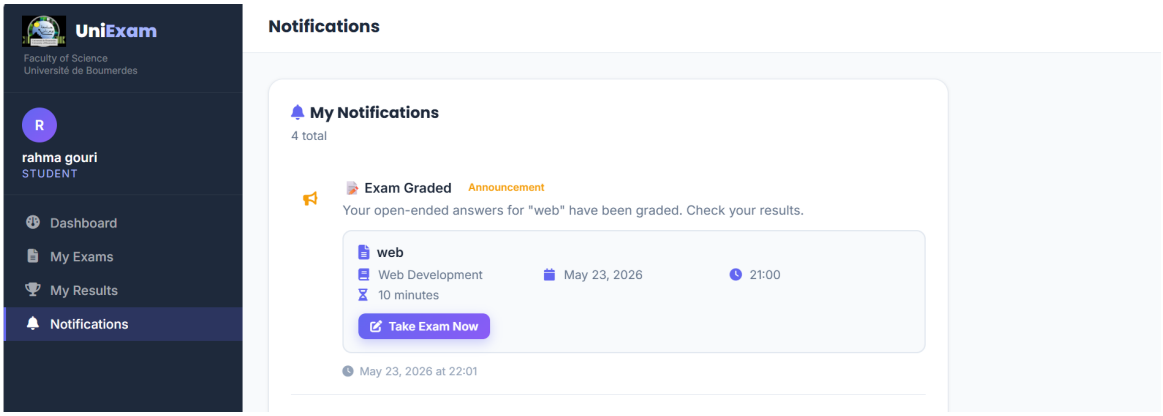


Figure 3.9: Student Taking Exam Interface.

Anti-Cheat Monitor

During an exam session, the Anti-Cheat Monitor runs silently in the background to ensure exam integrity. It works by listening to browser events such as tab switching, window blur, and fullscreen exit. Each detected violation increments a warning counter visible to the system, and once the maximum number of warnings is reached, the exam is automatically submitted regardless of completion status. All violations are timestamped and recorded in a cheating log stored alongside the submission, allowing the professor to review suspicious behavior after the exam. Additionally, the monitor enforces fullscreen mode at the start of the session and prevents access to browser developer tools,making it significantly harder for students to seek external help during the exam.

3.6 Conclusion

In conclusion, this chapter presented the practical implementation of the online exam management platform, bringing the previous design phases to life. We outlined the hardware and software development environment, followed by the justification of our DBMS selection to ensure robust data persistence and security. Finally, the showcase of the core user interfaces demonstrated that the platform achieves an intuitive, responsive, and functional design tailored to our university's needs. With these modules successfully developed and visualized, the implementation phase of the UniExam platform is now fully accomplished, delivering a functional solution ready to fulfill its objectives.

General Conclusion

This Bachelor's thesis marked the culmination of a comprehensive project aimed at modernizing academic workflows through the development of the UniExam platform. At the outset, our objective was to address the complexities of managing university examinations by designing a reliable, secure, and user-friendly digital system.

To achieve this, our work was structured into three complementary chapters:

- **Chapter 1** initiated our study with a comprehensive presentation of online examination systems, analyzing existing solutions to establish a clear framework and identify the specific requirements of our platform.
- **Chapter 2** transitioned into the formal design phase, utilizing UML (Unified Modeling Language) to construct the structural and behavioral architecture of the system through precise diagrams.
- **Chapter 3** brought this conceptual model to life through practical implementation. By selecting an appropriate development environment and a robust DBMS, we successfully translated our UML designs into a functional platform, validated by intuitive graphical user interfaces.

Beyond the academic requirements, this project has been an invaluable professional and personal experience. It allowed us to bridge the gap between software engineering theory and practice, mastering modern development tools while solving real-world logistical challenges within a university context.

While the current version of UniExam is fully operational and meets our initial goals, it also lays a strong foundation for future enhancements. Moving forward, the platform could be expanded by incorporating:

- **Advanced Security Modules:** Integrating biometric authentication or advanced anti-cheating mechanisms during online sessions.
- **Mobile Application Integration:** Developing a dedicated mobile version to complement the web platform for real-time push notifications.
- **AI-Driven Analytics:** Implementing automated statistics to analyze student performance trends over semesters.

Ultimately, we hope that UniExam serves as a valuable step toward the digital transformation of university administration, improving efficiency and accessibility for the entire academic community.

Webography

[1] World history encyclopedia –"the civil service examinations of imperial china"– updated on 08 february 2019 – visited at: 28 february 2026 –available at :. https://www.worldhistory.org/article/1335/the-civil-service-examinations-of-imperial-china/.

[2] Mercer | mettl glossary –online exam – last updated: No clear update date available – visited at: 28 february 2026 – available at :. https://mettl.com/glossary/o/online-exam/.

[3] Smowl – understanding online tests: function, purpose, and pros and cons – last updated: 16 june 2022 – visited at: 28 february 2026– available at :. https://smowl.net/en/blog/online-tests-definition-advantages/elementor-toc$_{heading-anchor-3}$.

[4] Bentayeb, n. (n.d.). testing and evaluation: Semester 1 course content (master 1 lla) [syllabus]. department of english language and literature, university batna 2– visited at: 28 february 2026 – available at :. https://acrobat.adobe.com/id/urn:aaid:sc:EU:2f48ffdf-4679-4c16-a3b3-b0f4492e3e19.

[5] Bentayeb, n. (n.d.). testing and evaluation: Semester 1 course content (master 1 lla) [syllabus]. department of english language and literature, university batna 2– visited at: 28 february 2026 – available at :. https://acrobat.adobe.com/id/urn:aaid:sc:EU:2f48ffdf-4679-4c16-a3b3-b0f4492e3e19.

[6] Moodle – the moodle story – last updated: No clear update date available– visited at: 6th mars 2026 – available at :. https://moodle.com/about/the-moodle-story/.

[7] bouncewatch – examonline – last updated: No clear update date available– visited at: 6th mars 2026 – available at :. https://bouncewatch.com/company/examonline.

[8] Formassembly – googleforms – last updated: No clear update date available– visited at: 6th mars 2026 – available at :. https://www.formassembly.com/glossary/google-forms/.

[9] Infonet – companies page – last updated: No clear update date available– visited at: 6th mars 2026 – available at :. https://infonet.fr/entreprises/79766249100019-inducido/.

[10] Evalbox – features page – last updated: No clear update date available– visited at: 25th may 2026 – available at :. https://evalbox.com/about.

[11] Meazurelearning – proctoru celebrates its 10th anniversary – last updated: October 23, 2018– visited at: 6th mars 2026 – available at :. https://www.meazurelearning.com/resources/proctoru-celebrates-10th-anniversary.

[12] Eu-startups – testwe – last updated: No clear update date available– visited at: 6th mars 2026 – available at :. https://www.eu-startups.com/directory/testwe/.

[13] Geeksforgeeks – use case diagram – last updated: No clear update date available– visited at: 27 mars 2026 – available at :. https://www.geeksforgeeks.org/system-design/use-case-diagram/.

[14] Figma – ressource library – last updated: No clear update date available– visited at: 27 mars 2026 – available at :. https://www.figma.com/resource-library/what-is-a-sequence-diagram/.

[15] creately – what is a uml class diagram – last updated: 02 february 2026 –by: Nuwan perera– visited at: 27 no clear update date available 2026 – available at :. https://creately.com/guides/what-is-a-class-diagram/.

[16] Github – plantuml – last updated: No clear update date available– visited at: 20 may 2026 – available at :. https://github.com/plantuml/plantuml.

[17] Drawio.com – about drawio – last updated: No clear update date available– visited at: 20 may 2026 – available at :. https://www.drawio.com/about.

[18] Infoworld – development tools – last updated: july,8th,2022– visited at: 20 may 2026– available at :. https://www.infoworld.com/article/2335960/what-is-visual-studio-code-microsofts-extensible-code-editor.html.

[19] Elementor blog – ressources – last updated: jan,2nd,2026– visited at: 20 may 2026– available at :. https://elementor.com/blog/what-is-wamp/.

[20] Adrienne watt.database design – 2nd edition:chapter 15 sql structured query language. bccampus– visited at: 21 may 2026– available at :. https://opentextbc.ca/dbdesign01/chapter/sql-structured-query-language/.

[21] Ovhcloud – what is html? – last updated: No clear update date available– visited at: 21 may 2026 – available at :. https://www.ovhcloud.com/en/learn/what-is-html/.

[22] Coursera – what is css? – last updated: feb,21th,2026– visited at: 23 may 2026– available at :. https://www.coursera.org/articles/css.

[23] Php – getting started – last updated: No clear update date available– visited at: 20 may 2026 – available at :. https://www.php.net/manual/en/introduction.php.

[24] Ovhcloud – what is javascript? – last updated: No clear update date available– visited at: 20 may 2026 – available at :. https://www.ovhcloud.com/en/learn/what-is-javascript/.

[25] Luis guerra – what is phpmyadmin? a comprehensive guide – last updated: No clear update date available– visited at: 20 may 2026 – available at :. https://www.luisguerra.co/en/what-is-phpmyadmin/.