Thesis submitted to the

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA, ALGERIA

1985

جامعة محمد بوضياف - المسيلة

Université Mohamed Boudiaf - M'sila

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

in Partial Fulfillment of the Requirements for the Degree of:

Master's in Computer Science

Specialization: Business Intelligence & Optimization

By

KHATIM Kenza Souha

DERNANI Somia

Entitled

DEVELOPMENT OF A WEB APPLICATION FOR ATTENDANCE TRACKING AND LECTURE DISTRIBUTION

Under the supervision of

Raouf Ouanis Lakehal Ayat

Publicly defended on 7th of June 2026

Jury Members

First name Last name	University of M'sila	President
First name Last name	University of M'sila	Reporter
First name Last name	University of M'sila	Examiner

الملخص

تتناول هذه المذكرة تصميم وتطوير تطبيق ويب لتتبّع الحضور وتوزيع المقررات في الوسط الجامعي. فالطرق اليدوية التقليدية، كأوراق التوقيع والجداول المطبوعة، بطيئة وعرضة للأخطاء وللحضور بالوكالة وصعبة التحديث. ولمعالجة هذه القيود، يوحّد التطبيق المقترح العمليتين ضمن منصّة واحدة قائمة على الأدوار تخدم المسؤول والأستاذ والطالب. ويُلتقَط الحضور بموثوقية عبر رمز QR يولّده الأستاذ مقروناً بالتحقّق من الموقع عبر GPS لمنع التحايل، في حين يدير توزيع المقررات الأساتذةَ والأقسامَ والجلساتِ والإسناداتِ دون تعارض. وقد نُمذِج النظام باستخدام مخطّطات UML ونُفِّذ بتقنيات ويب حديثة حول نموذج بيانات علائقيّ متماسك. والنتيجة أداة مركزية آنية تقلّل الجهد الإداري وتحسّن الدقة وتتوافق مع التنظيمات الجزائرية المتعلّقة بالحضور.

الكلمات المفتاحية:تطبيق ويب، تتبّع الحضور، توزيع المقررات، رمز ، نظام تحديد المواقع .

Abstract

This dissertation presents the design and development of a web application for attendance tracking and course distribution in a university setting. Traditional manual methods, such as paper sign-in sheets and printed timetables, are slow, error-prone, vulnerable to proxy attendance, and difficult to update. To address these limitations, the proposed application unifies both processes within a single, role-based platform serving administrators, teachers, and students. Attendance is captured reliably through a teacher-generated QR code combined with GPS location verification to prevent fraud, while course distribution manages teachers, classes, sessions, and assignments without conflicts. The system was modeled using UML diagrams and implemented with modern web technologies around a coherent relational data model. The result is a centralized, real-time tool that reduces administrative effort, improves accuracy, and aligns with Algerian university regulations on attendance.

Keywords - Examples: web application, attendance tracking, course distribution, QR code, GPS.

Dedication

To those who inherited the patience of Asiya, the loyalty of Khadija, the wisdom of Aisha, and the strength of Fatima… To the hearts overflowing with mercy and tenderness, who became our refuge in times of hardship and our light in moments of darkness.

*To my beloved mother, **Abdeli Rafika**, the fountain of endless love, the source of comfort and serenity… To you who sacrificed your peace for our happiness, stayed awake through countless nights for our well-being, and surrounded our lives with prayers, patience, and unconditional love. You have always been the greatest blessing bestowed upon us.*

*To my dear father, **Mahfoud Dernani**, the symbol of dignity, perseverance, and honor… To you who taught us that success is born from struggle, and that true strength is built through patience and hard work. You have been our guide, our support, and the example we proudly follow in life.*

*To my late grandmother, **Abdeli Zineb**, may Allah shower her with His infinite mercy and grant her the highest place in Paradise… Though she is no longer with us, her kindness, warmth, and beautiful memories remain forever alive within our hearts. May her soul rest in eternal peace.*

To our beloved siblings, and to every soul who taught us even a single letter, guided us with sincerity, or extended a helping hand along our journey… To everyone who believed in us, supported us, and shared both our joys and struggles, your kindness will forever remain engraved upon our hearts.

To each and every one of you, we dedicate this humble work with profound love, gratitude, and everlasting appreciation.

somia dernani

To those who inherited the patience of Asiya, the loyalty of Khadija, the wisdom of Aisha, and the fortitude of Fatima… to those whose hearts overflow with compassion, vaster than the mightiest rivers on Earth.

To you, the luminous guide of our path, the sanctuary of our journey, and the longing of hearts aflame with devotion… To you, the quenchless thirst of the yearning soul… To the noblest mothers this world has known.

To those who taught us the endurance to learn, the tenacity to succeed, and the resilience to persevere you have been the beacon of our purpose and the essence of our achievements.

To those who reign sovereign over the realms of our hearts… To our beloved parents, who nurtured us with boundless care and shared with us life's sweetness and trials alike.

To our cherished siblings, and to every soul who illuminated our minds with even a single letter guiding us, step by step, from childhood's classroom to the halls of academia.

To all who loved us unconditionally, who offered the gifts of kinship and unwavering kindness though words fail to name each of you, know that your presence is etched indelibly upon our hearts.

To each of you, we dedicate this humble endeavor.

Khatim kenza souha

Acknowledgements

All praise and gratitude belong to Allah, the Almighty, the Omnipotent, the First and the Last, the Manifest and the Hidden, who has bestowed upon us innumerable blessings, sustained us with His boundless provisions, and illuminated our path. To Him alone is due all praise and the utmost gratitude.He, the Most High, granted us the greatest of blessings by sending His servant and Messenger, Muhammad (peace and blessings be upon him), who conveyed the clear Qur'an, imparted knowledge of what we did not know, and inspired us to seek wisdom in all its forms.We praise and thank Him for granting us the strength, patience, and perseverance to overcome the challenges faced in completing this humble endeavor.We also express our profound appreciation to every educator who has enriched our intellect from the earliest stages of learning to the present. Our gratitude extends to all who have offered their support, near or far.We are particularly indebted to the faculty and staff of the Department of Computer Science at Mohamed Boudiaf University of M'sila for their invaluable guidance and assistance.In particular, we extend our heartfelt thanks to all the professors of the Computer Science Department, whose dedication and efforts have greatly contributed to our academic growth. We are also especially grateful to the Dean of the Faculty and the Head of the Department for their continuous support and encouragement.A special expression of sincere appreciation goes to our supervisor, **Dr. Raouf Ouanis Lakehal Ayat**, whose valuable guidance, patience, and insightful feedback have played a pivotal role in the success of this project.Finally, we humbly beseech Allah Almighty to bless us with righteousness, wisdom, moral fortitude, and contentment, and to count us among those who are rightly guided and who guide others.

Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

- **AI**: Artificial Intelligence
- **ML**: Machine Learning
- **NLP**: Natural Language Processing

Introduction

1. General Context

The gradual digitalisation of administrative processes represents a notable advance in contemporary institutional governance, enhancing efficiency, transparency and the quality of the services delivered to users. In the higher-education sector in particular, academic management is becoming increasingly complex as student numbers grow and programmes diversify. Within this landscape, two processes stand out for their direct impact on the quality of teaching and on the daily experience of students, teachers and administrators alike: the tracking of student attendance and the distribution of courses across teachers, groups, rooms and time slots.

For decades, these two processes were carried out manually, through sign-in sheets circulated in the classroom and printed timetables posted on notice boards. Despite the simplicity of this practice, it reveals growing operational limitations: it consumes teaching time, it is prone to human error during transcription and tabulation, it is vulnerable to fraud through signing on behalf of absent colleagues (proxy attendance), it makes archiving and review difficult, and it delays the delivery of updates to students whenever a timetable changes.

Web applications emerge as an innovative solution capable of addressing these challenges, since they centralise data in a single database, provide real-time, role-based access from any browser or device, and automate the capture of attendance, the calculation of attendance rates and the generation of absentee lists. This direction is reinforced in Algeria by an explicit national digitalisation policy, embodied in the higher-education digital platform (PROGRES) and the national digital master plan; indeed, the regulations in force (Ministerial Decision No. 1165 of 2025) explicitly mandate the digital recording of absences and their entry through the digital platform, which makes the development of a dedicated application for attendance tracking and course distribution both a practical need and a regulatory requirement at the same time.

2. Problem Statement

Despite the clear advantages, designing and deploying a unified application for attendance tracking and course distribution requires careful study of multifaceted challenges that go beyond mere technical implementation. This study is framed around the following main research question:

> *How can a web application be systematically designed and developed to unify attendance tracking and course distribution, so as to improve usability, accuracy and operational efficiency while complying with the regulatory requirements of the Algerian university?*

This central inquiry gives rise to the following sub-questions:

- How can attendance be captured accurately and reliably while resisting fraud (proxy attendance), through mechanisms such as QR codes and geographic location verification?
- How can the course-distribution functions and timetable management be designed so as to avoid conflicts between teachers, groups and rooms while reducing manual effort?
- How can the university's organisational structure (administrator, teacher, student) be translated into a tiered system of roles and permissions that respects Algerian regulations?
- What mechanisms can enable the production of transparent, real-time attendance reports, together with absentee and exclusion lists, accessible to all relevant stakeholders?
- How can the system's scalability and reliability be ensured so as to maintain consistent performance during the intensive-usage periods typical of the start of class sessions?
- Which evaluation methodologies are most appropriate for assessing usability and user satisfaction in the context of academic-management applications?

3. Objectives

This dissertation aims to design and implement a web application for attendance tracking and course distribution that prioritises user-centred design and operational robustness. The specific objectives are:

- To develop an intuitive, accessible user interface that allows the different roles (administrator, teacher and student) to navigate and perform their tasks smoothly.

- To implement a reliable, fraud-resistant attendance-capture mechanism that combines a QR code generated live by the teacher with location verification through the external GPS system.
- To design a course-distribution and timetable-management module that avoids conflicts, reduces manual work and allows rapid republishing whenever a change is made.
- To implement transparent mechanisms for aggregating and publishing attendance data and for producing absentee and exclusion lists, in line with the regulatory requirements of the Algerian university.
- To adopt rigorous software-engineering practices to ensure the platform's reliability, scalability and responsiveness under realistic operating conditions.
- To unify attendance tracking and course distribution within a single, coherent data model that eliminates duplication and preserves consistency.
- To carry out empirical usability testing and iterative improvement based on user feedback.

4. Structure of the Dissertation

This study is divided into three main chapters:

- **Chapter 1:** A general overview of attendance tracking and course distribution systems (the state of the art).
- **Chapter 2:** The design of our platform, with a particular focus on its modelling using UML (Unified Modeling Language) diagrams.
- **Chapter 3:** A description of the tools used to implement the platform, followed by a detailed account of its practical application with examples of the main interfaces developed.

Chapter 1

Attendance Tracking and Course Distribution Systems

1.1 Introduction

The management of academic activities in higher education institutions has grown increasingly complex as student populations expand and as universities are expected to deliver services that are fast, transparent and accurate. Among the many administrative processes a university must handle, two stand out for their direct impact on the quality of teaching and on the daily experience of students and staff alike: the tracking of student attendance and the distribution of courses across teachers, groups, rooms and time slots. For decades these tasks were performed manually, on paper and through informal arrangements, which made them slow, error-prone and difficult to audit.

This situation is now changing rapidly under the pressure of a broader digital transformation. National authorities increasingly treat the digitalisation of academic management not as a convenience but as a strategic priority; Algeria, for example, has adopted a digital master plan and has progressively integrated dozens of digital service platforms into a single higher-education information system, with the explicit aims of improving efficiency, reducing paper and coordinating services for students, teachers and administrators [1]. Within this movement, the web application has emerged as the natural delivery model.

A web application is well suited to academic management because it requires no installation on the user side, can be reached from desktops, laptops and smartphones alike, centralises data in a single database, and cleanly separates the user interface, the front end, from the data and business logic, the back end, hosted on a server [2]. The architectural conventions that make such applications scalable and maintainable, including the resource-oriented, stateless client-server style now dominant on the web, are themselves the product of sustained engineering research [3]. This chapter establishes the state of the art that frames the present work. It begins by tracing the evolution of academic management systems from paper records to integrated web platforms. It then examines, in turn, attendance tracking and course distribution, defining each concept, justifying its importance, contrasting traditional and digital practice, and reviewing the principal families of existing solutions. The chapter situates these processes within the organisational structure of universities and within the specific regulatory environment of Algerian higher education. Finally, it surveys international experience, identifies the recurring challenges reported by researchers and practitioners, and outlines the solutions that motivate the design choices made in the chapters that follow.

1.2 The Evolution of Academic Management Systems

Academic management has passed through several distinct stages, summarised in Figure 1.1. In the earliest stage every record was kept on paper: attendance was collected by circulating a signed sheet during the lecture, timetables were typed and posted on notice boards, and registration was handled at a physical counter. The approach is intuitive but fragile. Records are easily lost or damaged, consolidating them at the end of a semester is laborious, and the information they contain is available to only a few people in one place at a time [4].

A second stage introduced standalone desktop software. Spreadsheets and local database applications reduced some of the manual effort and improved legibility, but the data remained trapped on individual machines, was difficult to share between departments, and offered no remote access. Storing institutional data in a properly designed database, rather than in scattered files, was itself an important advance, because a database management system provides controlled, concurrent and consistent access to shared data while shielding applications from the physical details of storage [5].

The decisive change came with the third stage, in which the web became the dominant delivery model. Web-based systems store information centrally and expose it through a browser, so that administrators, teachers and students can each consult and update the data appropriate to their role from wherever they happen to be [6--8]. This model also made it natural to distinguish categories of user and to grant each of them different rights over the same shared data, a separation of roles that recurs throughout the systems reviewed in this chapter. The current stage is one of integration and automation. Institutions increasingly rely on student information systems, learning management systems and dedicated modules for scheduling and attendance that exchange data with one another rather than operating in isolation. The literature shows a clear trend towards combining a web platform with complementary technologies, whether biometric sensors, radio-frequency identification or mobile applications, so that data capture is automatic and the resulting information feeds directly into reporting and decision-making [7, 8]. Algeria's national PROGRES platform, which centralises the academic life of students within a single ministry-run information system, is a concrete embodiment of this integration stage [1]. The application proposed in this thesis belongs to the same stage: a single web application that unifies attendance tracking and course distribution within one coherent data model.

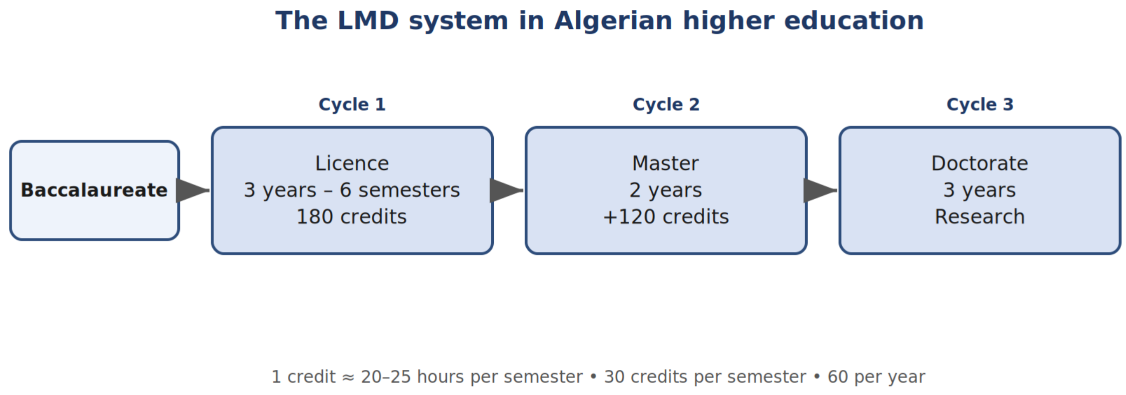

Figure 1.1: Evolution of academic management systems from paper records to integrated platforms.

1.3 Attendance Tracking

1.3.1 Definition of Attendance Tracking

Attendance tracking is the process of recording, storing and verifying the presence of students at scheduled pedagogical activities such as lectures, tutorials, laboratory sessions and workshops. A complete attendance system associates each presence or absence with a specific student, a specific session and a precise date and time, and makes this information available for later consultation and analysis. It is useful to distinguish attendance monitoring, the act of capturing whether a student is present, from attendance management, the broader handling of that data, including its aggregation, the calculation of attendance rates, the generation of absentee lists and the triggering of administrative consequences. Viewed as an information process, attendance tracking has three core components, shown in Figure 1.2. The first is an input mechanism that captures a piece of information unique to each student. The second is a processing layer that interprets and validates that information, for example by matching it against enrolled students and against the scheduled session. The third is a storage and presentation layer that records the result durably and displays it to authorised users in an appropriate form [4]. The whole forms a data lifecycle that begins with capture and ends with reporting, and the value of a system depends on how reliably it carries information through every stage of that lifecycle.

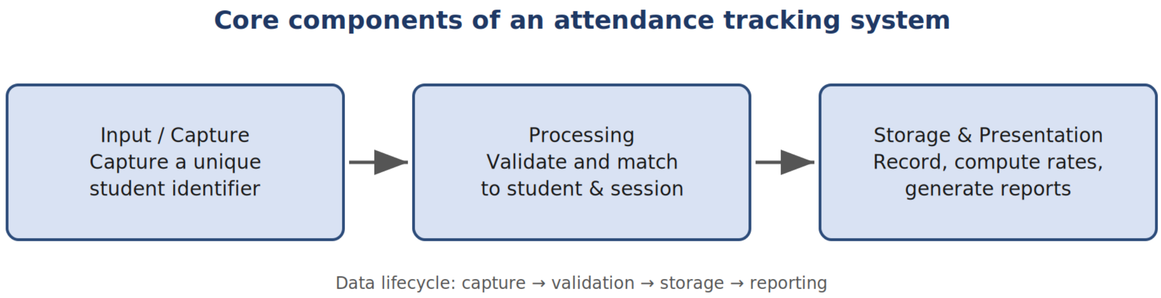

Figure 1.2: The three core components and data lifecycle of an attendance tracking system.

1.3.2 The Importance of Attendance Tracking

Interest in accurate attendance data is not merely administrative. A frequently cited large-scale study found a significant positive relationship between how regularly students attend and the marks they ultimately achieve, suggesting that attendance is a meaningful early indicator of academic engagement and risk [9]. Other authors have linked weak attendance to poor retention, framing attendance data as an instrument for identifying students who may be drifting away from their studies before they formally withdraw [10]. These findings give attendance data a value that extends well beyond compliance. For teachers, attendance records reveal patterns of behaviour and allow at-risk students to be identified while intervention is still possible. For administrators, consolidated attendance information supports operational and strategic decisions and supplies the evidence needed to apply institutional regulations fairly [1, 6]. At the level of the whole institution, the same data feed the statistics that universities and supervising ministries use to monitor the functioning of their programmes. Reliable attendance information therefore serves several stakeholders simultaneously: it protects the integrity of assessment by ensuring that only students who genuinely took part are credited, it supports communication with students and, where relevant, their families, and it underpins both pedagogical and administrative decision-making.

1.3.3 Traditional vs. Digital Attendance Methods

In the traditional method the teacher either calls out names or circulates a sheet of paper for students to sign. Although simple and requiring no equipment, this practice has welldocumented weaknesses. It consumes valuable teaching time, and for a class of around fifty students the roll call alone can take several minutes that are lost from instruction [4]. It is prone to human error during transcription, and it is vulnerable to fraud, since a present student can sign on behalf of an absent colleague, a practice known as proxy attendance. The same vulnerability affects automated but token-based methods, where the swapping of identification cards, sometimes called buddy punching, defeats the purpose of the system. Finally, paper records are difficult to archive, to search and to relate to earlier sessions [7]. Digital attendance methods address these shortcomings by automating capture and centralising storage. Once a session is opened, presence is recorded electronically and is immediately available for reporting, for the automatic calculation of attendance percentages and for the generation of absentee lists. Digital systems reduce the administrative burden, minimise transcription errors and make tampering more difficult, while allowing teachers and administrators to consult the data remotely [6, 8]. The principal trade-off is a dependence on supporting infrastructure, such as devices, sensors or a network connection, which must be considered carefully during design. In short, the move from traditional to digital attendance is not simply a change of medium but a change in what becomes possible: real-time access, automatic computation, durable archiving and resistance to a range of common abuses

1.3.4 Types of Attendance Tracking Systems

Although attendance systems share a common foundation, almost all of them communicate over an internet-protocol infrastructure, they differ markedly according to the identification technology used to recognise the student, and these differences translate into distinct profiles of cost, speed, accuracy and reliability [4, 6--8]. The principal families reported in the literature are summarised in Table 1.1.

Identification technology	Main advantages	Main limitations
RFID cards	Fast reading, high accuracy and reliability	Equipment and maintenance cost; cards can be swapped, lost or forgotten (proxy attendance)
Bluetooth	Low cost, low power, reasonably reliable	Limited operating distance; depends on device addresses that change frequently
Barcode / QR code	Inexpensive, accurate, easy to deploy	Slower registration; codes can be shared, copied or photographed
Fingerprint biometrics	Strong uniqueness, hard to forge, cannot be lost	Slower per-user capture; sensitive personal data raises privacy concerns; sensitive to dirt or injury
Face recognition	Contactless and fast under ideal conditions	Accuracy degrades with lighting, camera angle and appearance; privacy and algorithmic-bias concerns
Voice recognition	Simple to use, no extra hardware	Low recognition accuracy; serial (one-by-one) registration is slow
Pure web technology	High customizability, low cost, reachable from any browser or personal device	Depends on network availability; needs safeguards against remote (off-site) registration

Table 1.1: Comparison of identification technologies used in attendance tracking systems

Radio-frequency identification (RFID) is among the most widely deployed approaches; a representative example uses ultra-high-frequency RFID tags read by a single-board computer, with the captured events transmitted wirelessly to a web application that lets faculty, students and the department head monitor attendance and even generate a seating plan [7]. Biometric solutions are equally prominent. Fingerprint-based systems pair a capacitive sensor with a microcontroller and a web back end, so that a student is recognised through a one-to-many template match before the corresponding record is written to the database [4], whereas facial recognition systems compare a live camera feed against stored facial descriptors and may add a location check to discourage fraudulent marking [8]. Mobile approaches exploit the smartphones that students already carry, using technologies such as near-field communication, quick-response (QR) codes and satellite positioning to confirm presence without the institution having to purchase dedicated hardware [11]. Purely web-based systems go further still and dispense with special identification hardware altogether: the teacher opens a session in the browser and students confirm their presence from a personal device, an approach valued for its low cost and high customisability but one that must guard against off-site registration [6]. Regardless of the capture method, these systems share the same overall structure, illustrated in Figure 1.3: client devices used by different roles communicate over the web with a server that holds the application logic and a central database. A recurring conclusion of the literature is that no single technology dominates on every criterion; the appropriate choice depends on the institution's budget, its tolerance for error, its privacy obligations and the quality of its network, which is why hybrid designs that combine a web platform with a chosen capture method are increasingly common.

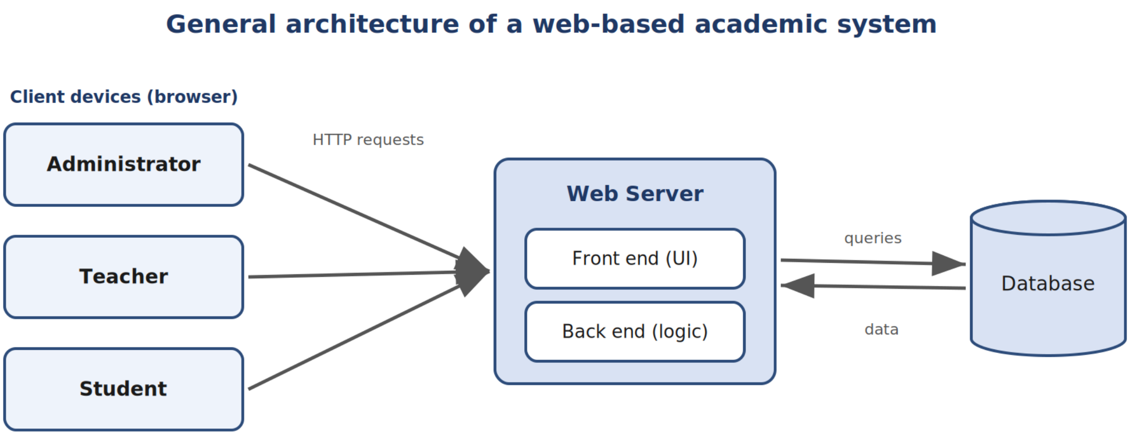

Figure 1.3: General architecture of a web-based academic system, with role-specific clients, a web server and a central database.

1.4 Course Distribution

1.4.1 Definition of Course Distribution

Course distribution, often referred to in the operational-research literature as university course timetabling, is the process of assigning the teaching events of a programme, that is, the pairing of courses with the teachers who will deliver them, to the limited resources available, namely student groups, rooms and time slots, in such a way that no conflict arises and that institutional rules are respected [12]. In a typical university the task therefore involves two related decisions: allocating courses to qualified teachers, and placing the resulting sessions on a weekly grid without clashes. Figure 1.4 represents this problem schematically, as the transformation of a set of resources into a feasible timetable under a set of constraints. The timetabling literature usually separates several variants of the problem. Class-teacher timetabling, characteristic of schools, assigns a fixed group its teachers; course timetabling, characteristic of universities, schedules courses that students choose individually; and examination timetabling schedules assessments under their own particular constraints [12, 13]. The present work is concerned chiefly with course timetabling in the university sense, where the same student may belong to several different course groups and where the avoidance of clashes between those groups is central.

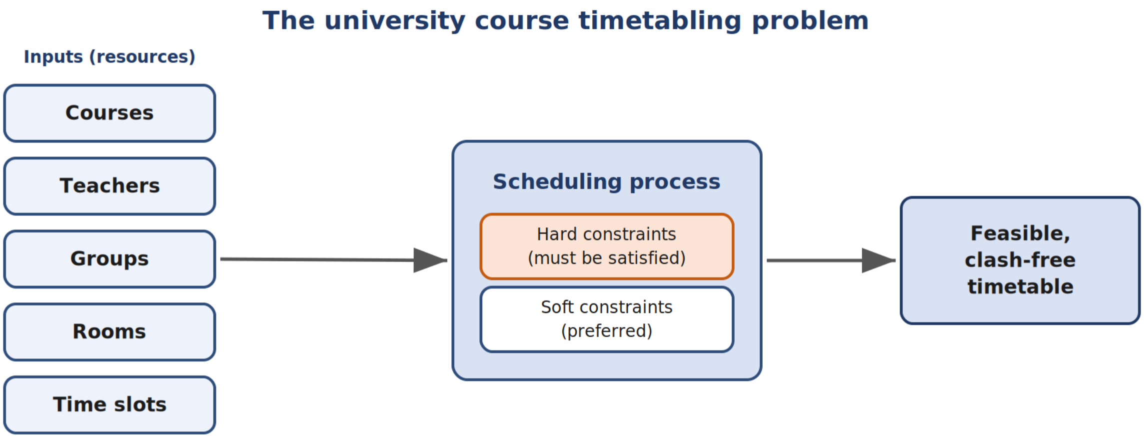

Figure 1.4: The university course timetabling problem: resources are scheduled into a feasible timetable subject to hard and soft constraints.

1.4.2 The Importance of Course Distribution

A well-constructed distribution has consequences that reach far beyond the administrator who prepares it. It determines whether rooms and laboratories are used efficiently, whether teachers' workloads are balanced and compatible with their availability, and whether students can follow all of the modules required by their curriculum without overlaps. Conversely, a poor distribution produces double-booked rooms, simultaneous sessions that a student cannot all attend, and uneven use of facilities, each of which degrades the quality of teaching and generates continuous administrative rework [12]. The importance of the task is amplified by its instability. Timetables frequently have to be revised during the year as staff change, as enrolments shift and as rooms become unavailable, and under a manual regime the cost of rescheduling and of communicating each change is itself a significant burden. A distribution that can be regenerated and republished quickly is therefore not a luxury but a practical necessity, which is one of the strongest arguments for moving the process onto an online platform.

1.4.3 Traditional Course Registration Methods

Traditionally, course distribution and the associated registration have been handled by hand. A member of staff, often a department head, builds the timetable on paper or in a spreadsheet, drawing on personal knowledge of the constraints, and the result is then printed and displayed on a notice board. Student registration in the course offering is similarly manual, carried out at a counter or through paper forms. This way of working is slow and laborious, it depends heavily on the individual who performs it, and the resulting documents are easy to misread, miscopy or forget. Beyond its inefficiency, the manual approach scales badly. As the number of courses, groups and rooms grows, the number of potential conflicts grows far faster, and a human scheduler can no longer be confident of having found a clash-free arrangement, let alone a good one. Any change forces the whole notice-board cycle to be repeated, so updates reach students late and inconsistencies between successive versions are common. These limitations motivated several decades of research into automating the construction of timetables.

1.4.4 Automated Course Distribution Systems

Automated course distribution is a long-standing research subject precisely because the problem is computationally hard. University timetabling belongs to the class of NP-complete (NP-hard) problems, which means that the effort required to find an optimal solution grows very rapidly with the number of courses, groups and rooms, so that exhaustive search becomes impractical for realistic instances [12, 13]. A foundational distinction in this field is between hard constraints, which must never be violated, such as a teacher being scheduled in two places at once, and soft constraints, which are desirable but not mandatory, such as avoiding undesirable time slots; a timetable is feasible when it satisfies all hard constraints, and the aim is then to satisfy as many soft constraints as possible [12]. The methods proposed over the years fall into a small number of families, summarised in Table 1.2. Early work applied exact operational-research techniques, including graph-colouring formulations, integer and linear programming, and constraint satisfaction, which can guarantee optimality on small problems but do not scale well [12, 13]. Constructive heuristics build a feasible timetable quickly by ordering events according to rules of thumb, and are often used to produce an initial solution. Metaheuristics then improve that solution: single-solution methods such as tabu search, simulated annealing and the great deluge algorithm refine one candidate using memory or relaxed acceptance rules, while population-based methods such as genetic and evolutionary algorithms, ant colony optimisation and particle swarm optimisation evolve a whole population of candidate timetables against a fitness function [12, 13].

Approach family	**Representative techniques**	**Characteristics**
Exact / operational research	Graph colouring; integer and linear programming; constraint satisfaction	Can guarantee optimal solutions on small instances but scale poorly as size grows
Constructive heuristics	Sequential and graph-colouring ordering rules	Build a feasible timetable quickly; final quality depends on the chosen ordering heuristic
Single-solution metaheuristics	Tabu search; simulated annealing; great deluge	Iteratively improve one solution, using memory or acceptance rules to escape local optima
Population-based metaheuristics	Genetic and evolutionary algorithms; ant colony; particle swarm	Evolve a population of candidate timetables against a fitness function expressing the constraints
Hybrid / service-oriented	Genetic algorithm with tabu search; multi-agent and web-service architectures	Combine the strengths of several methods and deliver scheduling as an online, integrated service

Table 1.2: Principal families of approaches to automated course timetabling

Comparative studies report that hybrid schemes, which combine the rapid feasibility of heuristics with the optimising power of metaheuristics, tend to outperform any single technique, and that memory-based and evolutionary searches in particular give strong results on standard benchmark instances [13]. The more recent and most relevant trend for this thesis, however, concerns delivery rather than the algorithm itself: scheduling logic is increasingly embedded in service-oriented or multi-agent web architectures, so that a timetable can be generated automatically, integrated directly into a wider university management system, and published online to students and staff [12, 13]. This trajectory mirrors the general evolution described in Section 1.2, from isolated single-purpose programs towards web-based, integrated and automated services, and it is precisely the niche that the proposed application occupies.

1.5 Academic Institutional Structure

1.5.1 Definition of Academic Institution

An academic institution, in the sense used in this thesis, is an organisation whose purpose is to provide higher education and to conduct research, and which is empowered to award recognised qualifications. Universities are the most prominent example, but the category also includes institutes, higher schools and university centres. What distinguishes such institutions, from the point of view of an information system, is the regular, cyclical nature of their activity: cohorts of students enrol, follow a structured programme organised into modules and semesters, are assessed, and progress or graduate, while the same teaching resources, teachers, rooms and time slots, are reused from one period to the next. This cyclical regularity is exactly what makes the activity amenable to systematic, software-supported management.

1.5.2 Organizational Structure of Universities

Universities are typically organised as a hierarchy, illustrated in Figure 1.5. At the top sits the central administration, the rectorate, headed by a rector and supported by vice-rectorates responsible for areas such as pedagogy, post-graduate studies, scientific research and external relations. Beneath this level the university is divided into faculties or institutes, each led by a dean and grouping cognate disciplines. Faculties are themselves subdivided into departments, which are the operational units closest to teaching and which are run by a department head assisted by section and specialisation coordinators.

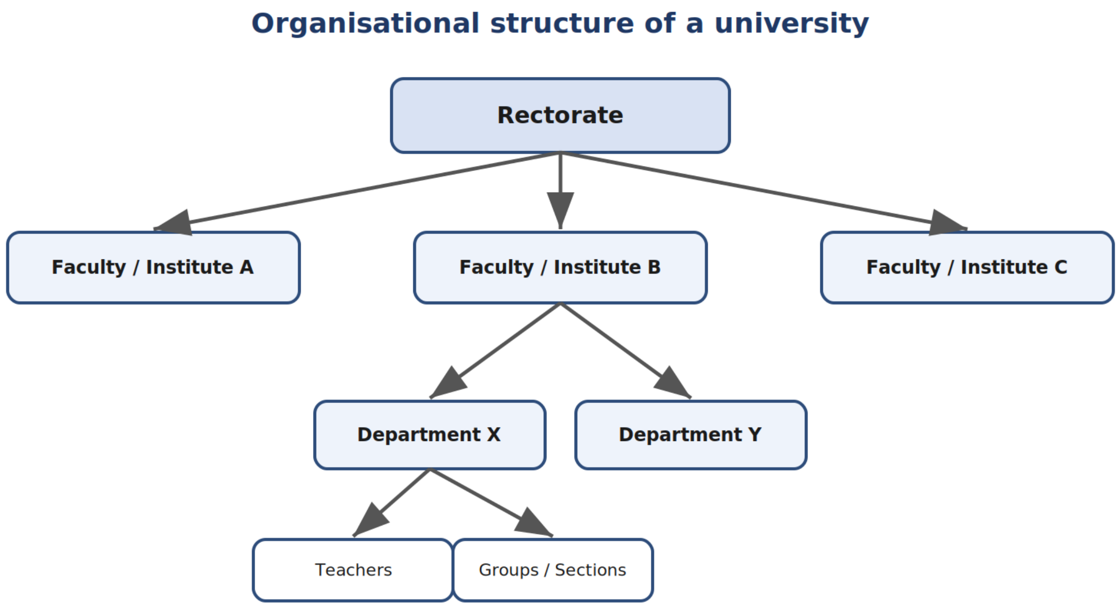

Figure 1.5: Typical hierarchical organisation of a university, from the rectorate down to teaching groups

This layered structure matters directly for the design of an academic management system, because authority, and therefore the right to read or modify particular data, is distributed across the levels. What an administrator may do differs from what a department head, a teacher or a student may do, and a faithful system must reproduce this distribution of responsibilities through a scheme of user roles and permissions. The role separation observed in the reviewed systems, where administrators, teachers and students each see a different view of the same data, is thus not an arbitrary design choice but a reflection of how universities actually function [6, 7].

1.5.3 Core Functions of Academic Departments

The department is where most of the processes addressed by this work actually take place. It is responsible for offering the modules of its programmes, for assigning those modules to its teachers in line with their specialisation, for organising students into groups and sections, and for building and maintaining the timetable. The department also oversees the day-to-day monitoring of teaching, which includes the supervision of attendance: it is at department level that absence reports are gathered and that the lists of students affected by attendance rules are prepared. Because attendance tracking and course distribution are both rooted in the department, an information system that serves the department directly is well placed to unify them. The teacher allocations and group arrangements produced when the department builds its timetable are the very same structures against which attendance is later recorded; managing them within one system therefore eliminates duplication and keeps the two processes consistent. This observation underlies the central design decision of the present thesis.

1.6 Educational Regulations in Algeria

1.6.1 Higher Education Laws and Policies

Algerian higher education is governed by a framework of laws and executive texts. The orientation law on higher education, Law No. 99-05 of 4 April 1999 [14], as subsequently amended and supplemented, defines the general organisation of the sector and established the three-degree architecture of Licence, Master and Doctorate for most disciplines, with certain fields such as medicine, pharmacy, dentistry, architecture, veterinary and agronomic sciences retaining longer single-cycle programmes [15]. The reform that introduced this architecture, widely known by its French acronym LMD, was enacted through Executive Decree No. 04-371 of 21 November 2004 [16] and has been applied progressively in Algerian universities since the 2004--2005 academic year, aligning the national system with the European Bologna Process [15]. More recently, the sector has pursued an explicit policy of digitalisation. Under a national digital master plan the ministry has integrated a growing number of digital service platforms into a single higher-education information system, intended to streamline academic management for students, teaching staff and administrators and to move the sector towards a paperless mode of operation [1]. The central instrument of this policy is the PROGRES platform, which manages the academic records of students nationwide. This regulatory and policy context is significant for the present work, because it shows that a web-based academic management application is not only technically desirable but actively encouraged by national policy.

1.6.2 Attendance Requirements in Algerian Universities

Attendance is treated by the Algerian regulatory framework as an essential component of academic evaluation rather than as an optional formality. Under the ministerial decision governing registration, the organisation of training and assessment in the Licence and Master tracks (Ministerial Decision No. 1165 of 4 October 2025), students are required to attend the various pedagogical activities, lectures, tutorials, laboratory work, workshops and field placements, regularly. The decision provides that accumulating five absences, whether justified or unjustified, in the tutorial, laboratory, workshop or field-placement sessions of a given module leads to exclusion from that module for the semester concerned; an excluded student may not sit any makeup examination in that module, and compensation between modules, units or semesters does not apply in such a case [17]. Crucially for the present work, the same decision specifies the operational mechanism by which attendance is to be monitored. Teachers are responsible for checking presence in each session and for preparing detailed absence reports that are transmitted to the department before the end-of-semester examinations, while the department head enters the absence data through the ministry's digital platform and draws up the lists of excluded students [17]. This explicit requirement for systematic, session-by-session capture and for the digital recording of attendance provides a direct regulatory justification for a dedicated attendance-tracking application, and it defines, in effect, several of the functional requirements that such an application must meet.

1.6.3 Course Registration Procedures (LMD System)

Within the LMD system, study is organised into semesters and measured in credits, as outlined in Figure 1.6. Each Licence comprises six semesters and corresponds to one hundred and eighty credits, while the Master adds a further one hundred and twenty credits over two years; a student capitalises thirty credits for each validated semester and sixty for each validated year [15]. Teaching is structured into teaching units, which group together component subjects delivered in different forms, lectures, tutorials, laboratory work and the like, and each subject carries a coefficient and is assessed on a scale of zero to twenty. A teaching unit is validated when its average reaches the pass mark, and credits, once obtained, are capitalisable and transferable from one course to another; validation may also be achieved by compensation between subjects or units, within the rules laid down by the regulation. Registration and re-registration in these tracks are themselves framed by ministerial decision, which sets out how students enrol in the course offering and progress through the cycle [17]. For a course-distribution module, these rules are important because they define the units, coefficients and group structures that any timetable and any registration record must respect; the regulation thus shapes the data model as much as the regulations on attendance shape the monitoring process.

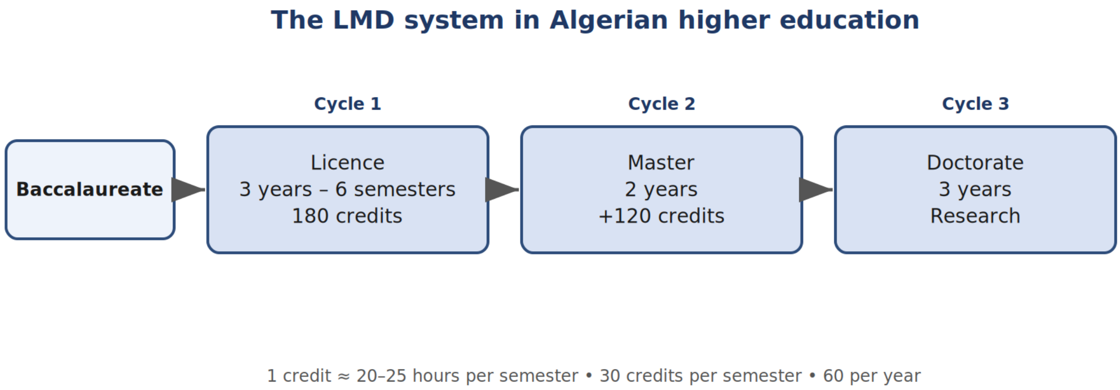

Figure 1.6: Typical hierarchical organisation of a university, from the Structure of the LMD system and the credit framework in Algerian higher education.

1.7 International Examples

1.7.1 Pioneering Countries in Academic Management Systems

The modernisation of academic management has advanced in parallel across many countries, often driven by broader reforms of higher education. In Europe, the harmonisation pursued through the Bologna Process encouraged institutions to standardise programmes and to adopt information systems capable of handling credits, mobility and comparable qualifications; it is this same movement that Algeria joined through the LMD reform [15]. Europe has also been the home of a sustained research community around timetabling, exemplified by the Practice and Theory of Automated Timetabling series and by international timetabling competitions, whose benchmark instances have allowed competing algorithms to be compared rigorously [12, 13]. Beyond Europe, universities in East and South-East Asia and in Africa have been particularly active in publishing prototypes that combine web platforms with identification hardware. The examples reviewed below, drawn from North Macedonia, Finland, the Philippines, Malaysia and Japan, illustrate both the geographical breadth of this activity and the diversity of the technical choices that institutions make when they set out to replace manual procedures with automated, data-driven ones.

1.7.2 Case Studies on Successful Implementations

Several documented implementations illustrate what such systems can achieve, and they are summarised together in Table 1.3. In North Macedonia, a web-based attendance solution built entirely with free and open-source technologies, HTML, CSS, JavaScript, PHP and MySQL, was deployed at a university and evaluated through a survey of teachers, students and managers; respondents reported that the system was easy to use, that registration was fast and unobtrusive, and that managers valued the ability to review all teaching activity for planning and decision-making [6]. In Finland, a bachelor's thesis project integrated a capacitive fingerprint sensor and a microcontroller with a web application built using a modern front-end framework and a document database, demonstrating accurate, real-time recording with substantially reduced administrative effort compared with the paper method [4]. In the Philippines, a classroom system combined ultra-high-frequency RFID with a motion sensor and a single-board computer, feeding a web application that allowed faculty, students and the chairperson to monitor attendance and to manage seating, explicitly with the goal of reducing opportunities for forgery [7]. In Malaysia, a real-time web-based system used facial recognition to display student details and an approximate location for each enrolled course, again as a means of preventing fake attendance [8]. In Japan, an earlier proposal showed that students' own mobile devices could be used together with a web application to register attendance, anticipating the now-common idea of reusing personal hardware to avoid dedicated equipment [11].

System / source	Country	Core technology	Distinctive feature
RENAP [6]	North Macedonia	Web only (HTML, CSS, JS, PHP, MySQL)	Open-source, fast registration on a local server
Savonia thesis [4]	Finland	Fingerprint sensor + microcontroller + web	Biometric capture with real-time web display
ReBICTE system [7]	Philippines	UHF RFID + single-board computer + web	Seating-plan generation and anti-forgery focus
WAS-FR [8]	Malaysia	Web + facial recognition	Live face matching combined with a location check
Mobile/web system [11]	Japan	Students' mobile devices + web app	Reuses personal devices, avoiding dedicated hardware

Table 1.3: Synthesis of representative web-based academic systems reviewed in this chapter.

Taken together, these cases confirm that web-based academic systems are technically feasible across very different settings, that they are generally well received by their users, and that they deliver measurable gains in speed, accuracy and administrative efficiency. They also show a common architectural pattern, a central database and server exposing role-specific views to administrators, teachers and students, that the present work adopts and extends.

1.7.3 Lessons Learned from Failures

The same body of work is candid about shortcomings, and these lessons are as instructive as the successes. Evaluations of biometric and recognition-based systems repeatedly note accuracy limitations; facial-recognition accuracy, for example, degrades sharply with changes in lighting, camera angle or appearance [8]. RFID and card-based approaches, although fast, remain vulnerable to card swapping and to students forgetting their cards. Purely web-based solutions, for their part, depend on network connectivity, so that a failure of the connection can block registration entirely, and they create the possibility of a student registering without being physically present unless additional safeguards are built in [6]. Two further lessons recur. First, commercial subscription platforms often lack the flexibility that individual institutions need, because requirements differ markedly from one country and one institution to another, which argues in favour of custom, locally adapted solutions. Second, several authors stress the importance of thorough early testing, since compatibility problems between hardware and software discovered late in a project are costly to resolve [4]. On the course-distribution side, the corresponding lesson from the optimisation literature is that algorithms validated only on simplified, artificial models often fail to transfer to real institutions, which underlines the value of testing any scheduling component against genuine institutional data [13]. These observations directly inform the design priorities adopted in this thesis: customisability, resilience, realistic validation and careful attention to the integrity of the captured data.

1.8 Challenges and Solutions

1.8.1 Common Challenges in Attendance Tracking

The literature reviewed above points to a consistent set of difficulties in attendance tracking. The first is fraud, principally proxy attendance, in which a present student records the presence of an absent one, whether by signing for a colleague, swapping an identification card or registering remotely. The second is accuracy and reliability, since biometric and recognition technologies are sensitive to environmental conditions and to the quality of the captured signal. The third is cost, both of acquiring specialised hardware and of maintaining it over time. The fourth is privacy and data protection, because biometric and personal data are sensitive information that must be stored securely and handled in conformity with the applicable rules.

The fifth is dependence on infrastructure, especially network connectivity, on which web-based capture relies [4, 6]. A sixth, more practical challenge is integration: an attendance system that cannot exchange data with the institution's registration and scheduling records duplicates effort and risks inconsistency.

1.8.2 Common Challenges in Course Distribution

Course distribution presents challenges of a different nature, rooted in combinatorial complexity. As an NP-hard problem it cannot, in general, be solved optimally by exhaustive search for realistic instances [12]. The numerous and sometimes conflicting constraints, the avoidance of clashes between teachers, groups and rooms, respect for teacher availability and room capacity, and the desire to satisfy soft preferences, make any manual solution laborious and any automated one demanding to design and to tune [12]. Frequent in-year changes compound the difficulty, since each modification can propagate clashes elsewhere and must then be recommunicated to everyone affected. A further challenge, again, is integration with the rest of the academic information system, so that the groups and units used in scheduling are exactly those used in registration and in attendance.

1.8.3 Proposed Solutions

In response to these challenges the recurring recommendation of the literature is to build a custom, web-based and integrated solution. A web architecture provides centralised, real-time, role-based access from ordinary browsers and devices, which simultaneously addresses remoteness, currency of information and infrastructure cost; adopting well-established architectural conventions for such applications helps to keep the result scalable and maintainable [2, 3]. For attendance, the risk of fraud can be reduced by combining session opening controlled by the teacher with appropriate verification, and by recording each event with its precise time so that anomalies become detectable; designing the system around free and widely supported technologies, rather than expensive proprietary hardware, keeps it affordable and customisable [6]. For course distribution, embedding heuristic and metaheuristic algorithms in the web platform allows feasible timetables to be generated quickly while leaving a human administrator in control of the soft preferences [12, 13]. The decisive insight, however, is that attendance tracking and course distribution are not independent problems but two faces of the same underlying data: the assignment of courses, teachers, groups, rooms and time slots produced by the distribution process is exactly the structure against which attendance is later recorded. Unifying them within a single web application, governed by one consistent data model and one scheme of user roles, therefore avoids duplication, removes the inconsistencies that arise when separate tools are used, and gives each category of user, administrator, teacher and student, a coherent view of the information that concerns them.

Sound software-engineering practice, in which requirements are analysed before the system is designed and the design is validated before deployment, then translates this insight into a dependable product [18, 19]. This unification is the central proposition of the present work.

1.9 Conclusion

This chapter has set out the conceptual, technical and regulatory background against which the proposed system is designed. It traced the evolution of academic management from paper records towards integrated web platforms, and it examined the two processes at the heart of this thesis. Attendance tracking was defined and justified by its documented link to academic performance and retention, the weaknesses of the traditional roll call were contrasted with the advantages of digital capture, and the principal identification technologies were compared. Course distribution was defined as a constrained, NP-hard assignment problem, its main variants were distinguished, and the move from manual timetabling towards automated, web-delivered solutions was reviewed across the major families of algorithms. The chapter then placed these processes within the organisational structure of universities and within the Algerian regulatory framework, which not only attaches strong consequences to attendance through the rules on exclusion but also explicitly mandates its digital recording and actively promotes the digitalisation of academic management. International case studies confirmed the feasibility and benefits of web-based academic systems and revealed a common role-based architecture, while the reported failures yielded clear design lessons concerning accuracy, cost, privacy, connectivity, realistic validation and the value of customisation. Finally, the analysis of challenges led to the central argument of this thesis: that attendance tracking and course distribution share the same data and are best addressed together, within a single web application. The next chapter builds on this foundation by analysing the requirements of such a system and specifying its functional and non-functional characteristics.

Chapter 2

Chapter 2: Desing

2.1 Introduction

In the previous chapter, we focused on studying attendance tracking and course distribution systems in higher education institutions. This study allowed us to analyze existing approaches, identify their strengths and limitations, and collect the information necessary to understand the requirements of the proposed solution. From this analysis, we extracted a set of functional and technical requirements that will guide the design of the future system, ensuring that it satisfies user needs and improves the management of attendance and course scheduling processes. At this stage, we will use UML (Unified Modeling Language) to model and describe the main components of the system design. The UMLStar tool will be used to create and represent the different diagrams of the proposed attendance tracking and course distribution system.

2.2 Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a standard language used to create the blueprints of software systems. UML can be used to visualize, specify, construct, and document the artifacts of a system with a strong software component [20].

2.2.1 The importance of using the Unified Modeling Language (UML)

- Enables professional software design.
- Allows for easy and low-cost modifications and maintenance.
- UML diagrams help developers quickly and easily understand the system.
- Acts as a communication language between developers and designers.

- Provides documentation of the design before programming begins.
- Is independent of any specific programming language or development process [20].

2.2.2 Types of UML Diagrams

In UML, there are 14 types of diagrams divided into two main categories:

- Structural Diagrams
- Behavioural Diagrams

As shown in Figure 2.1 [20]:

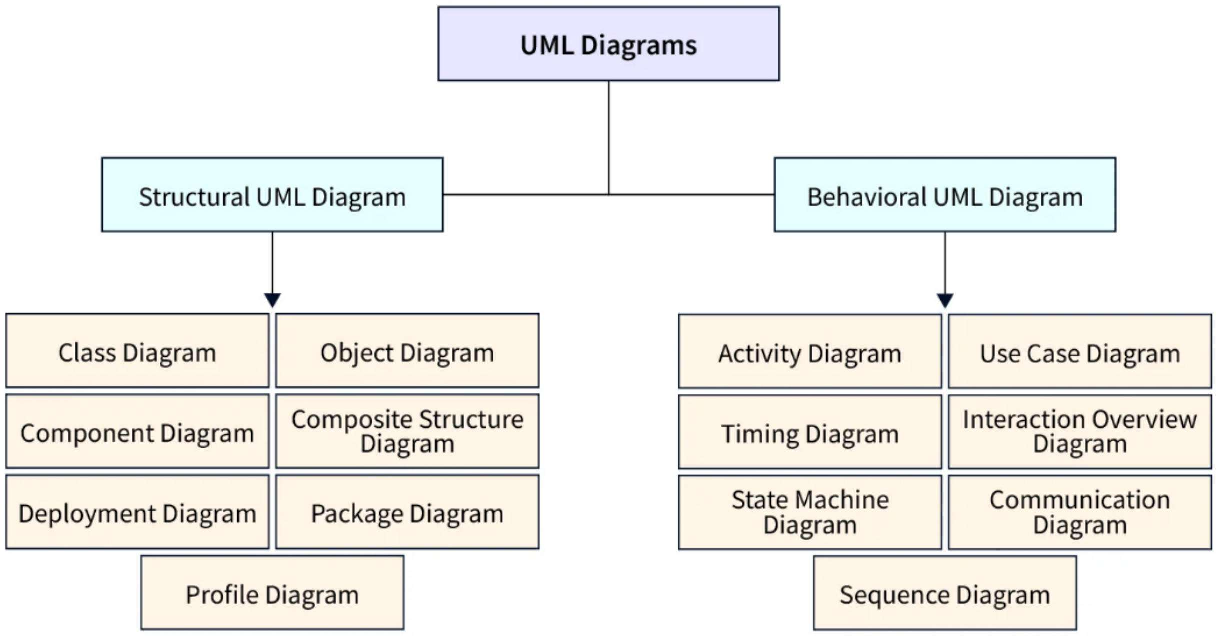

Figure 2.1: Types of UML Diagrams

2.3 Diagrams Used in the Design

Various types of diagrams are used in software modeling. In our project, we have chosen to use four key diagrams, which are listed below:

2.4 Use Case Diagram

The Use Case Diagram is employed to collect and describe the requirements of the system's stakeholders.It illustrates the interaction between the actors and the system by identifying the specific use cases and the corresponding actors responsible for executing them.This diagram is foundational in capturing functional requirements and provides a high-level overview of how external entities interact with the system.Two fundamental concepts underpin the use case diagram:

- Actor: An external user or another system that interacts with the system under development
- Use Case: A specific function or service that the system provides in response to an actor's interaction [18]

As illustrated in the figure:

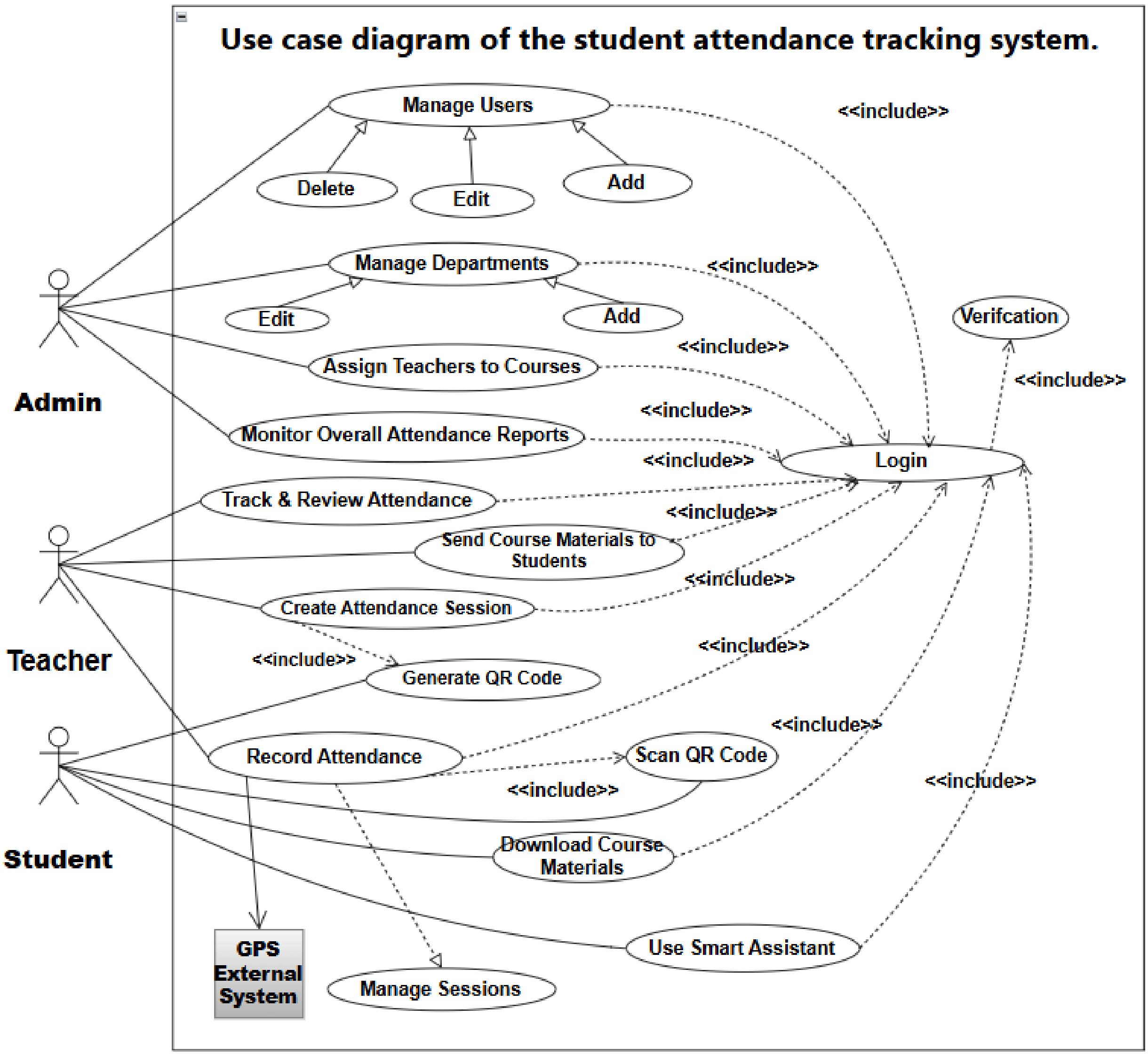

Figure 2.2: Use case diagram of the student attendance tracking system.

Discraption:

This diagram illustrates how three types of users interact with the various functions of the attendance tracking system.

Actors:

- **Admin**: Has the highest level of permission. Can manage users and departments, assign teachers to courses, create attendance sessions, generate QR codes, and monitor attendance reports.
- **Teacher**: Responsible for recording student attendance (using QR code scanning or other methods) and downloading course materials. Can also use a smart assistant.
- **Student**: Participates in attendance sessions (by scanning a QR code). The student has no management permissions.
- **GPS External System**: An external system that interacts with the attendance system to verify student locations during attendance recording (for example, to prevent cheating).

Use Cases:

Use Cases for Admin:

- **Manage Users**: Managing user accounts (students, teachers) including add, edit, and delete.
- **Manage Departments**: Managing academic departments including add and edit.
- **Assign Teachers to Courses**: Assigning teachers to specific courses.
- **Monitor Overall Attendance Reports**: Monitoring general attendance reports for all students.
- **Track & Review Attendance**: Tracking and reviewing student attendance records.
- **Send Course Materials to Students**: Sending educational materials to students.
- **Create Attendance Session**: Creating a new attendance session for a specific lecture.
- **Generate QR Code**: Generating a QR code for the attendance session that students will scan.

Use Cases for Teacher:

- **Record Attendance**: Recording student attendance.
- **Scan QR Code**: Scanning the QR code generated by the admin or system to verify student attendance.
- **Download Course Materials**: Downloading educational materials sent by the administration.
- **Use Smart Assistant**: Using a smart assistant (possibly for inquiries or reports).

Use Cases for Student:

- **Manage Sessions**: Managing their own sessions (such as attending lectures). The student relies on the external GPS system to record attendance.

Relationships:

Include Relationship:

The <<include>> relationship means that one use case cannot be completed without executing another use case.

- **Monitor Overall Attendance Reports** includes **Track & Review Attendance**: You cannot monitor general attendance reports without first tracking and reviewing attendance.
- **Assign Teachers to Courses** includes **Track & Review Attendance**: When assigning a teacher to a course, there must be a system to track their attendance.
- **Send Course Materials to Students** includes **Track & Review Attendance**: Sending materials requires knowing which students are present to receive them.
- **Create Attendance Session** includes **Generate QR Code**: Creating an attendance session requires generating a QR code for it.
- **Record Attendance** includes **Scan QR Code**: Recording attendance cannot happen without the student scanning the QR code.

2.4.1 Sequence Diagram

The sequence diagram is a type of interaction diagram within the Unified Modeling Language (UML) that captures the dynamic behaviour of a system by modelling the sequence of interactions between various structural elements over time. It visualizes the objects and classes involved in a particular scenario and the chronological order of message exchanges among them to realize specific system behavior. Sequence diagrams are typically linked to use cases as they provide a detailed logical representation of system execution. This diagram is also referred to as an event diagram or an event scenario in some contexts [21]

1) Sequence Diagram “Login”:

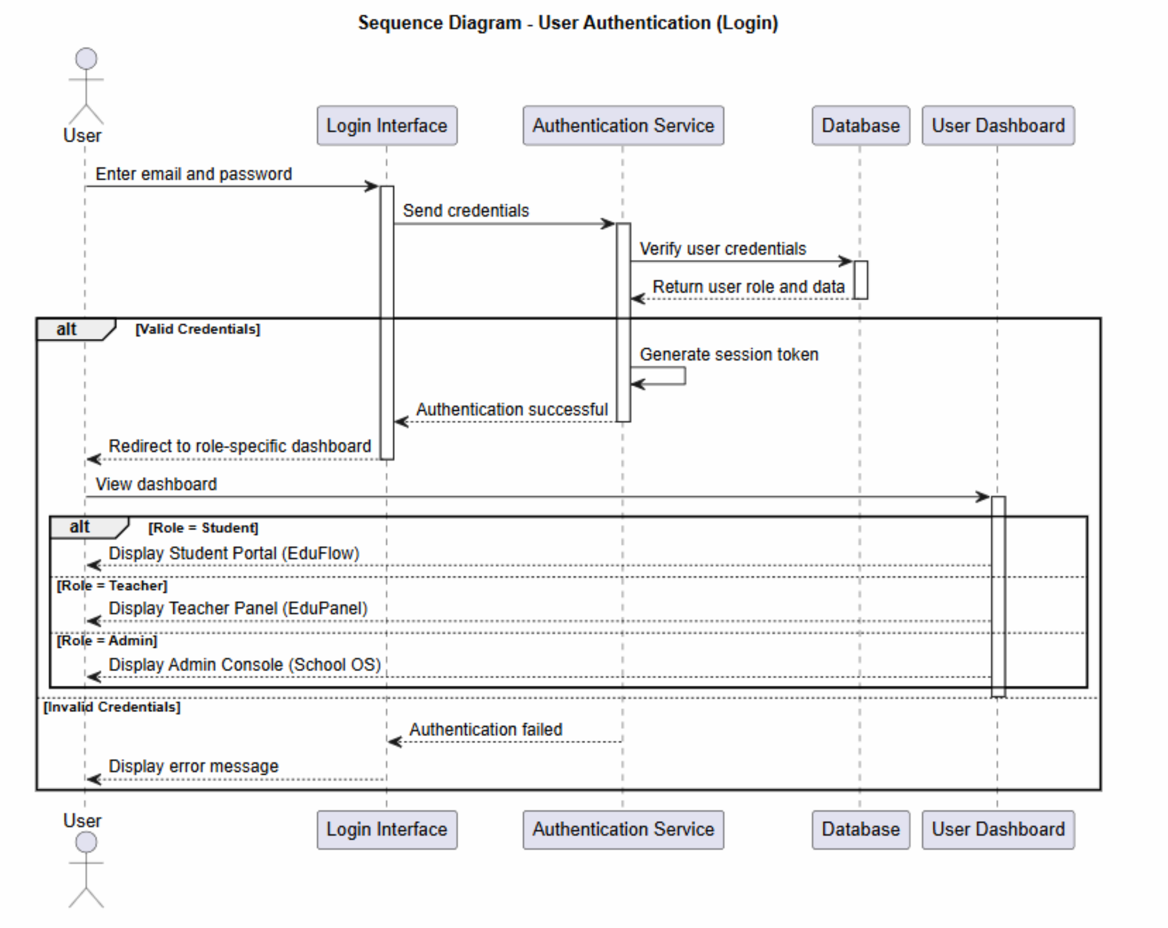

Figure 2.3: Sequence Diagram “Login"

Description:

This Sequence Diagram illustrates the user authentication (login) process in the School OS system. The user enters an email and password through the login interface, which sends the credentials to the authentication service. The service verifies the information with the database and retrieves the user role and account data. If the credentials are valid, the system generates a session token and redirects the user to the appropriate dashboard based on their role, such as Student Portal, Teacher Panel, or Admin Console. If the credentials are invalid, the system displays an authentication error message.

2) Sequence Diagram "Admin Dashboard and Quick Stats":

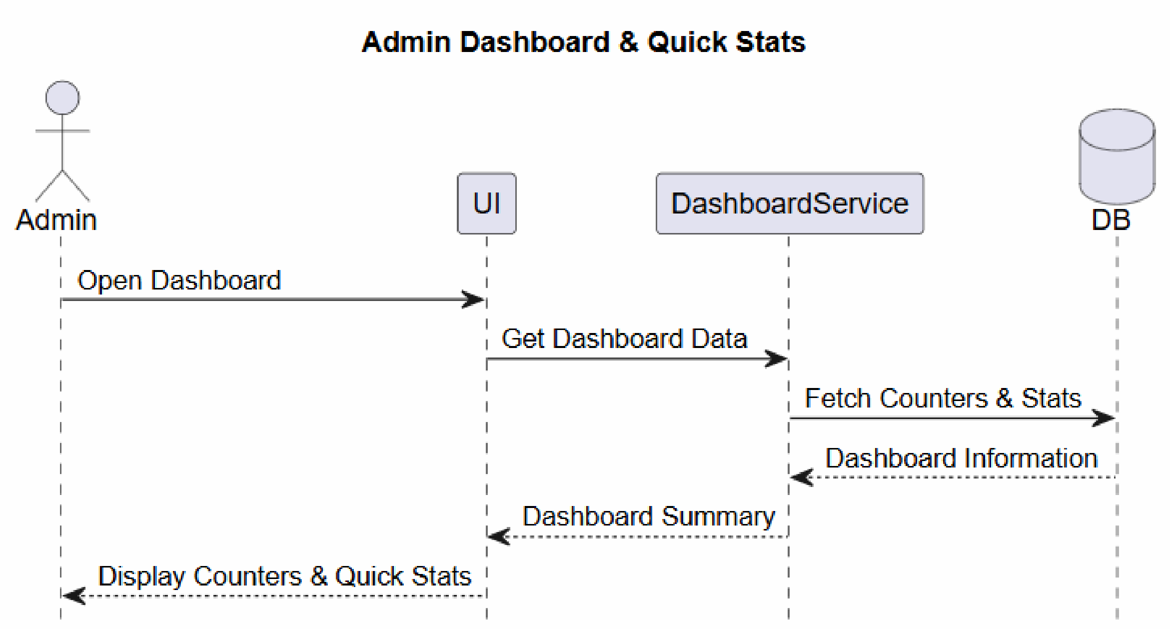

Figure 2.4: Sequence Diagram "Admin Dashboard and Quick Stats"

Description:

This Sequence Diagram describes how the admin dashboard displays system statistics. The dashboard service retrieves information such as the number of users, departments, classes, and sessions, as well as role-based statistics for students, teachers, and admins. The interface then renders these statistics as badges and progress indicators.

3) Sequence Diagram “Add New User”:

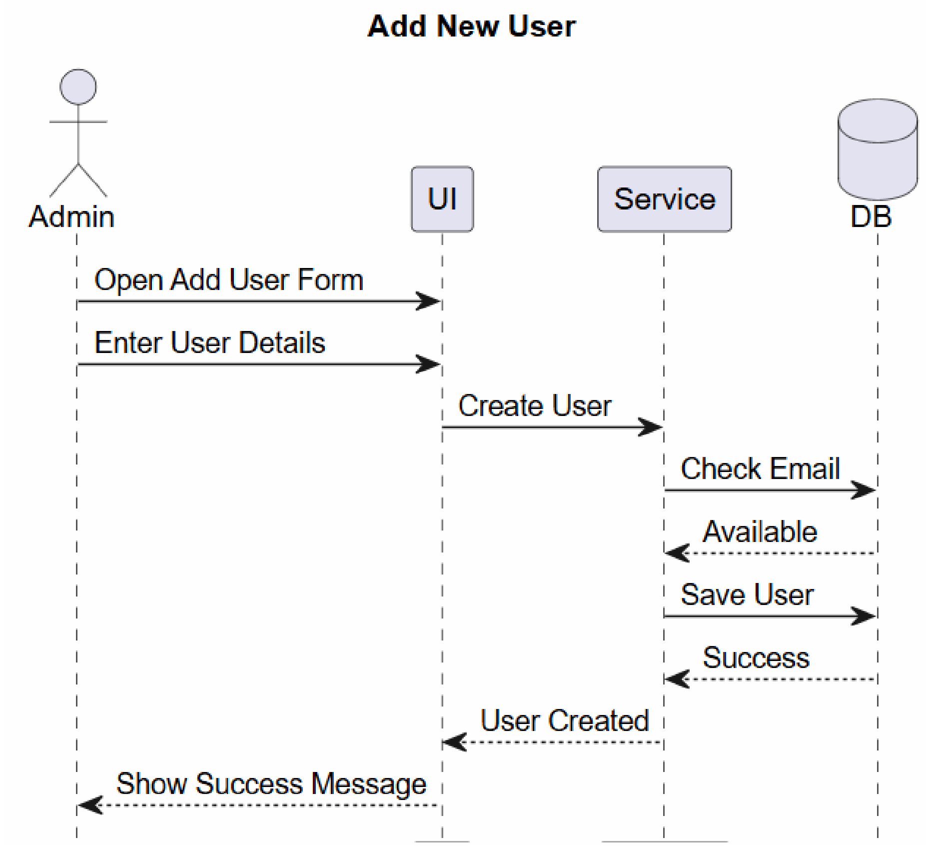

Figure 2.5: Sequence Diagram “Add New User"

Description:

This diagram represents the process of adding a new user to the system. The admin enters user information such as name, email, and role. The user service verifies that the email is not already registered, creates the account in the database, and updates the dashboard counters after a successful operation

4) Sequence Diagram “Create New Department”:

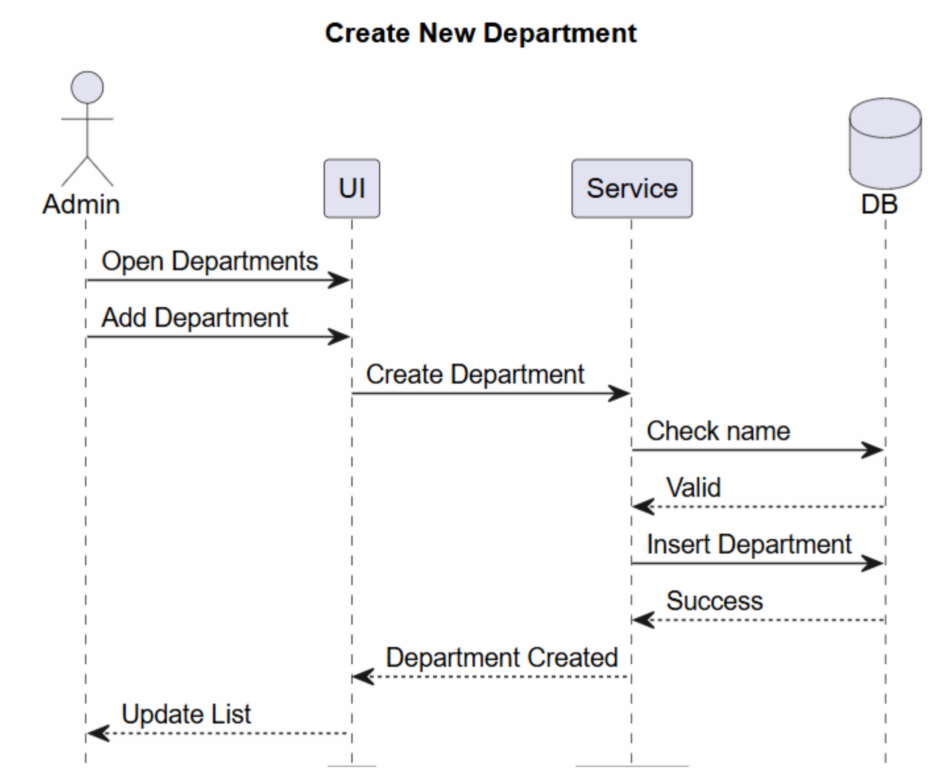

Figure 2.6: Sequence Diagram “Create New Department"

Description:

This Sequence Diagram shows how an administrator creates a new department in the system. The admin opens the departments page, enters the department name, and submits the form. The department service checks if the name already exists, then stores the new department in the database. Finally, the interface updates the department list and displays a success message.

5) Sequence Diagram “Create New Session”:

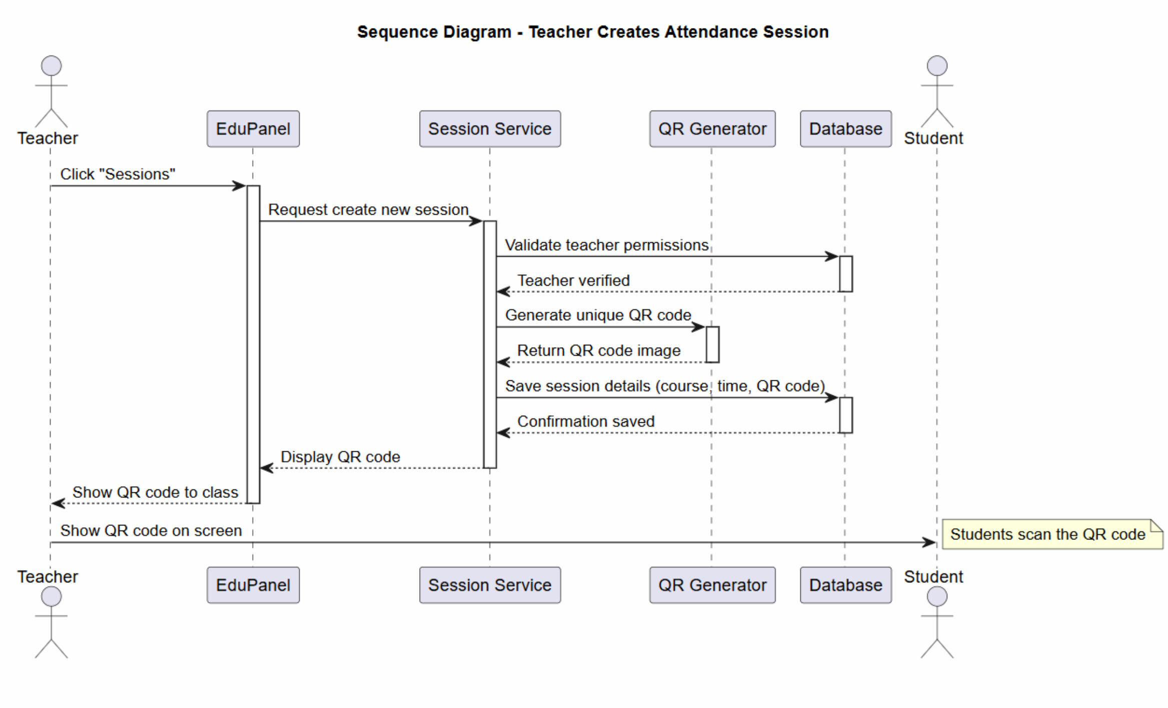

Figure 2.7: Sequence Diagram “Create New Session"

Description:

This diagram shows how an administrator creates a new session. The admin enters session information including the session name and schedule. The session service checks for time conflicts, then stores the session in the database. After successful creation, the sessions table and counters are updated.

6) Sequence Diagram “Assigning a teacher to a class or session”:

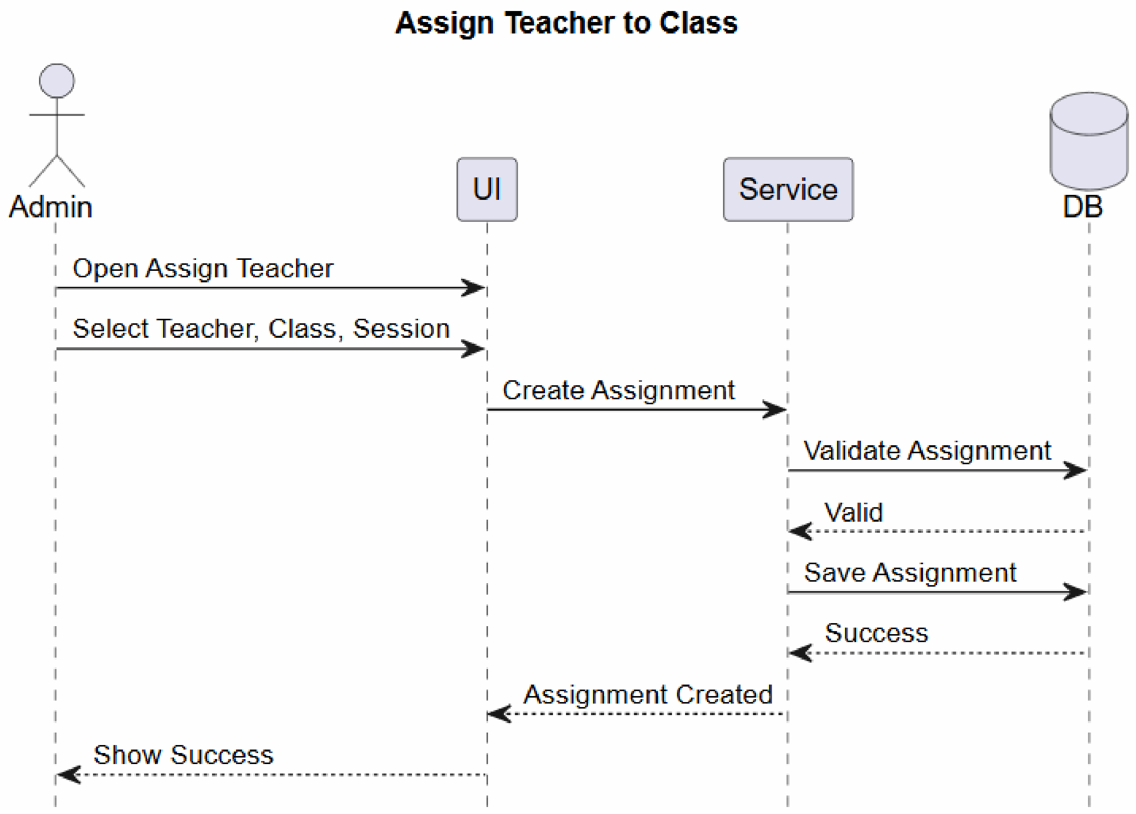

Figure 2.8: Sequence Diagram “Assigning a teacher to a class or session"

Description:

This diagram describes the process of assigning a teacher to a class or session. The admin selects a teacher, class, and session from the interface, then the assignment service validates the request to avoid duplicates. After validation, the assignment is saved in the database and the updated assignment list is displayed.

7) Sequence Diagram “Teacher Dashboard Overview”:

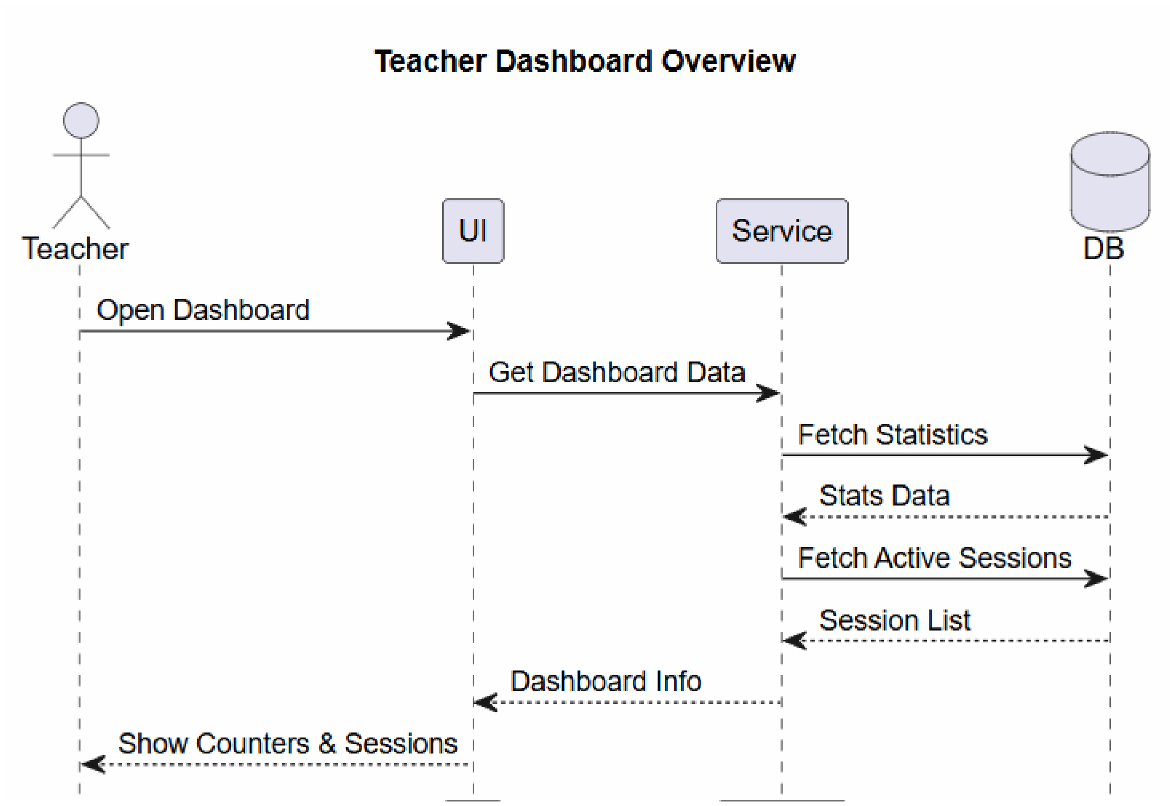

Figure 2.9: Sequence Diagram “Teacher Dashboard Overview"

Description:

This diagram describes how the teacher dashboard is loaded. When the teacher opens the dashboard, the system retrieves summary statistics such as sessions, students, classes, and quizzes from the database. The interface then displays these statistics along with active session cards.

8) Sequence Diagram “Generate Attendance QR Code”:

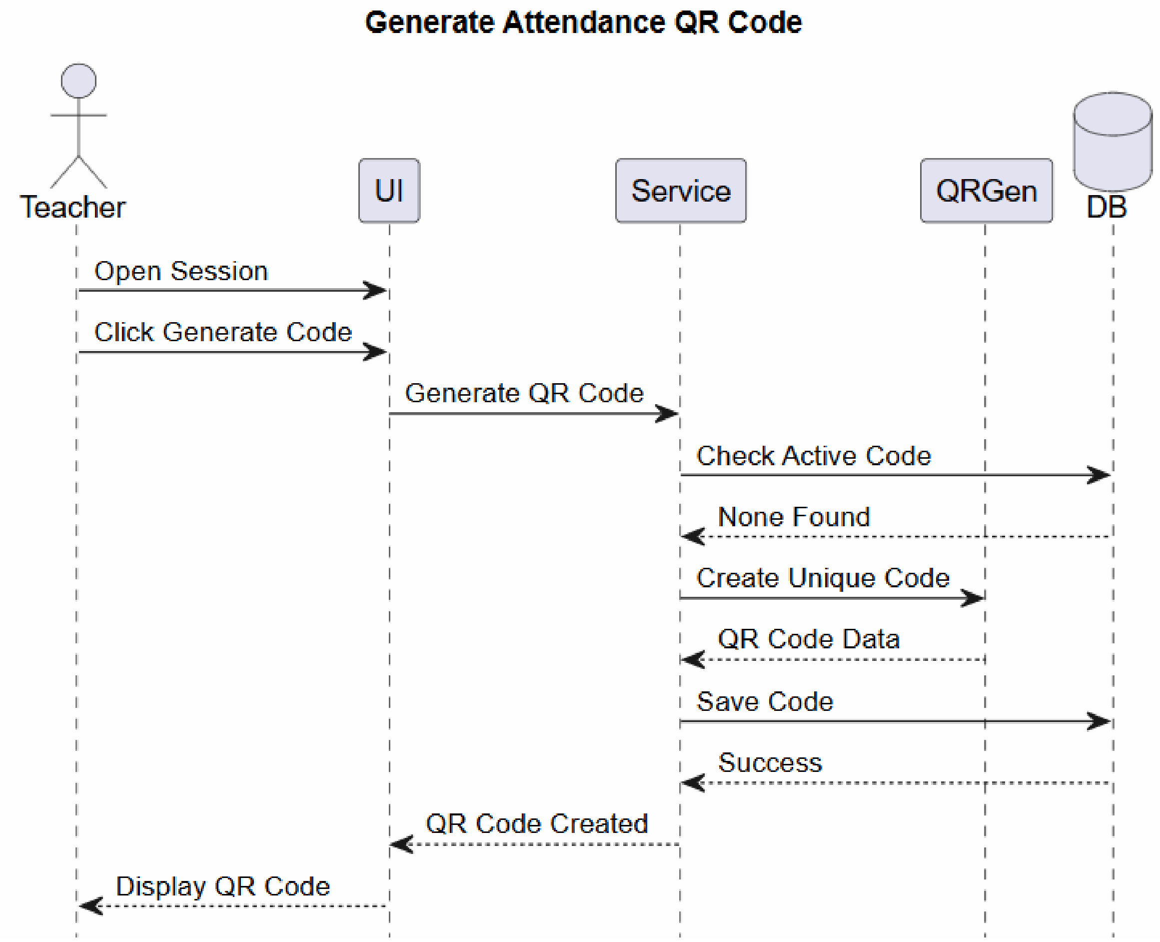

Figure 2.10: Sequence Diagram “Generate Attendance QR Code"

Description:

This diagram explains how a teacher generates a QR code for attendance tracking. The teacher opens a session and clicks the generate button. The attendance service verifies that no active code already exists, generates a unique QR code, saves it in the database, and sends it back to the interface for display.

9) Sequence Diagram “Add New Quiz”:

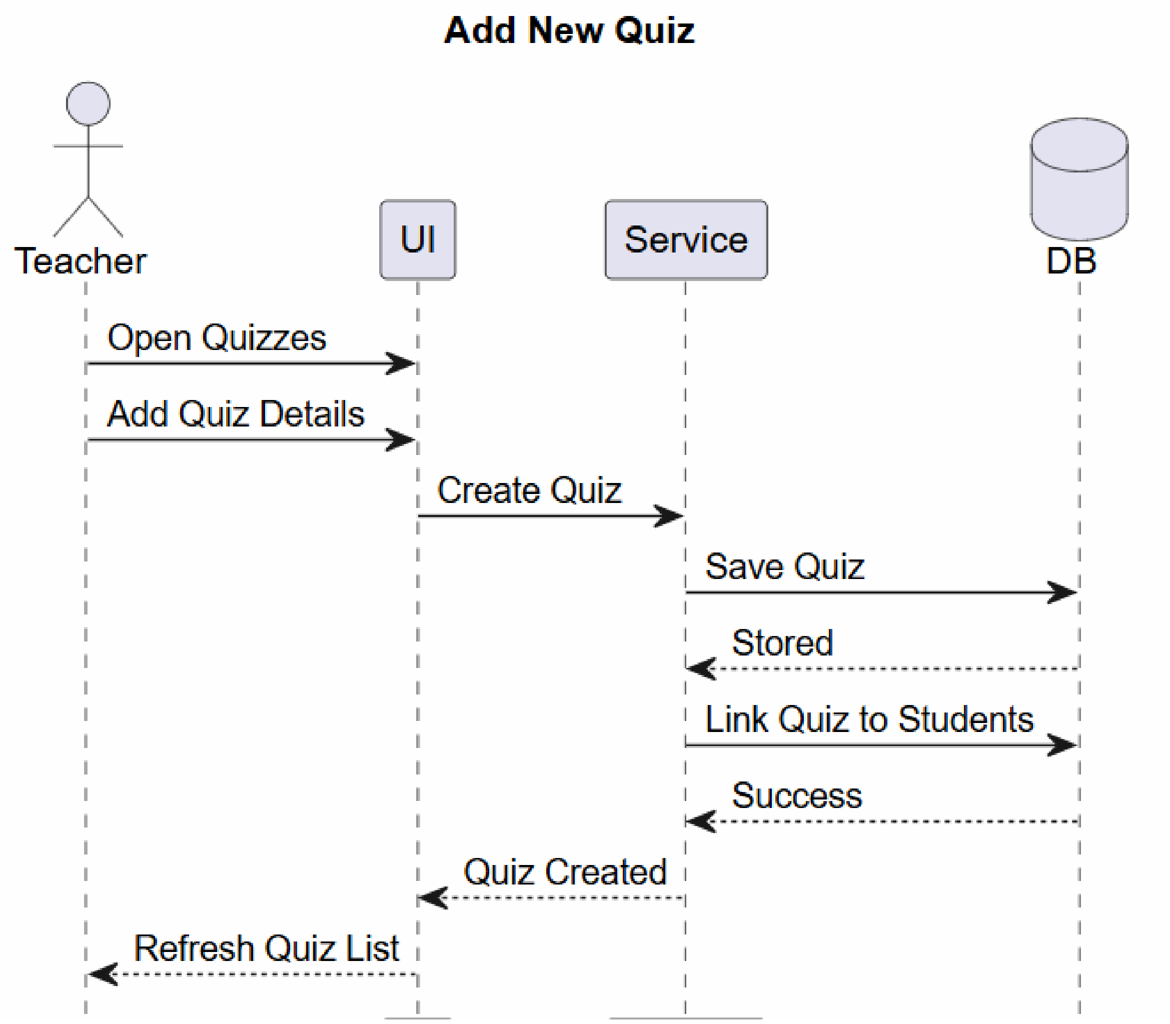

Figure 2.11: Sequence Diagram “Add New Quiz"

Description:

This Sequence Diagram explains how a teacher creates and publishes a new quiz. The teacher enters quiz details or uploads a file, and the quiz service stores the quiz information in the database. The quiz is then linked to the assigned students, and the interface refreshes the quizzes list.

10) Sequence Diagram “View Real-time Attendance and AI History”:

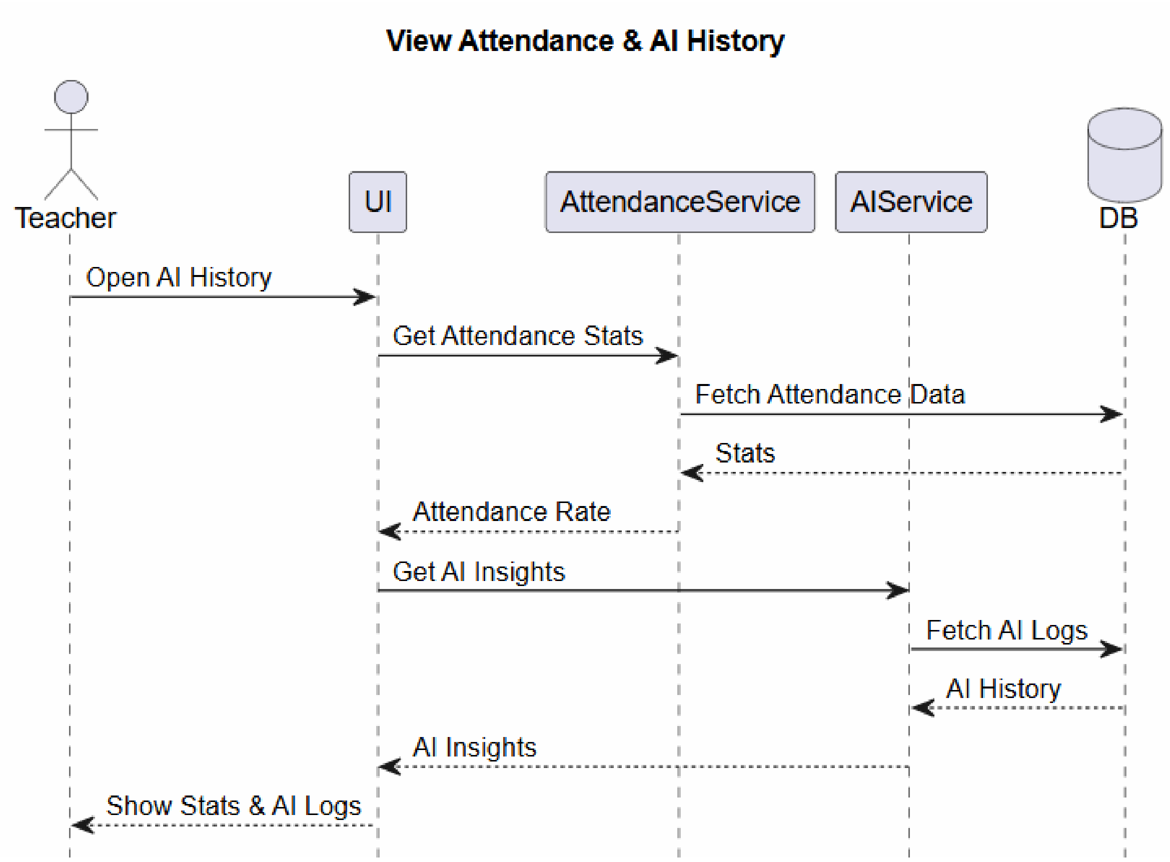

Figure 2.12: Sequence Diagram “View Real-time Attendance and AI History"

Description:

This diagram illustrates how teachers can monitor live attendance and AI analytics. The interface requests attendance statistics and AI insights for a specific session. The system retrieves attendance data and AI history logs from the database, then displays them on the dashboard in real time.

11) Sequence Diagram “Student Dashboard and Course Materials”:

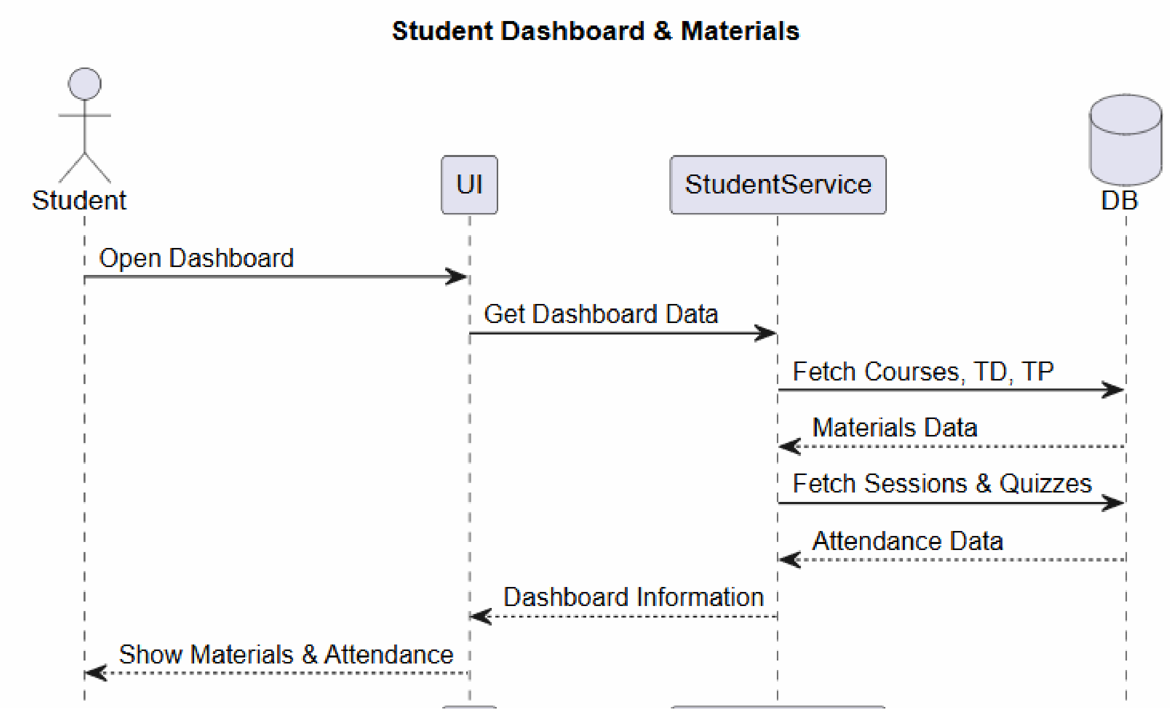

Figure 2.13: Sequence Diagram “Student Dashboard and Course Materials"

Description:

This diagram explains how the student dashboard displays academic resources and schedules. The student service retrieves courses, TDs,TPs, quizzes, and session information from the database. The interface then presents these details as dashboard counters and sidebar statistics.

12) Sequence Diagram "Validate Attendance by QR":

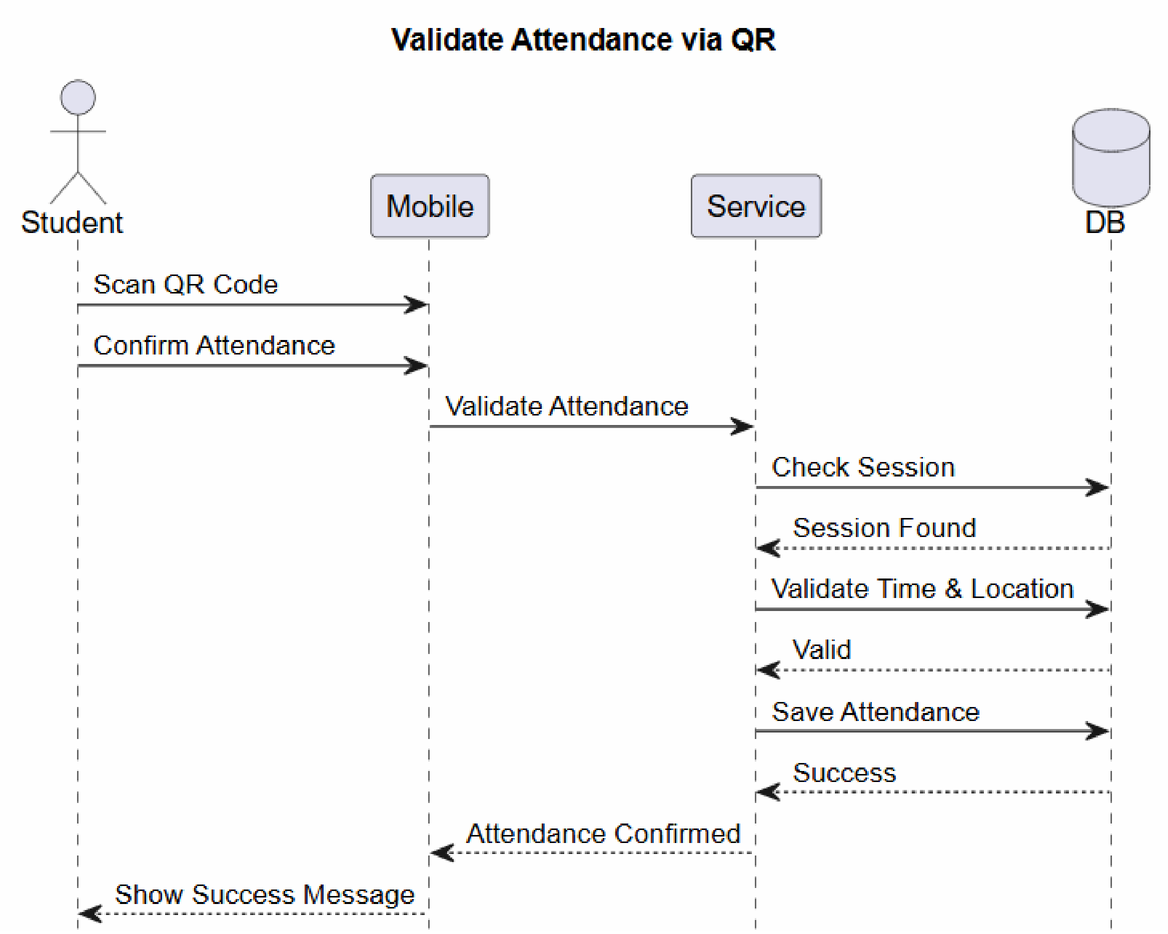

Figure 2.14: Sequence Diagram "Validate Attendance by QR"

Description:

This Sequence Diagram illustrates how a student validates attendance using a QR code. The student scans the QR code and confirms attendance through the mobile application. The validation service checks the session, time, and location before recording attendance and updating session statistics. A success message is then shown to the student.

2.4.2 Class Diagram

The class diagram is among the most fundamental and widely used diagrams in UML. It offers a static structural view of a system by depicting its classes and the relationships among them, including associations such as composition, aggregation, and inheritance. Each class in the diagram is characterized by three key elements:

- The Class name

- The Attributes
- The Operations [18].

Figure below presents the class diagram of the student attendance tracking:

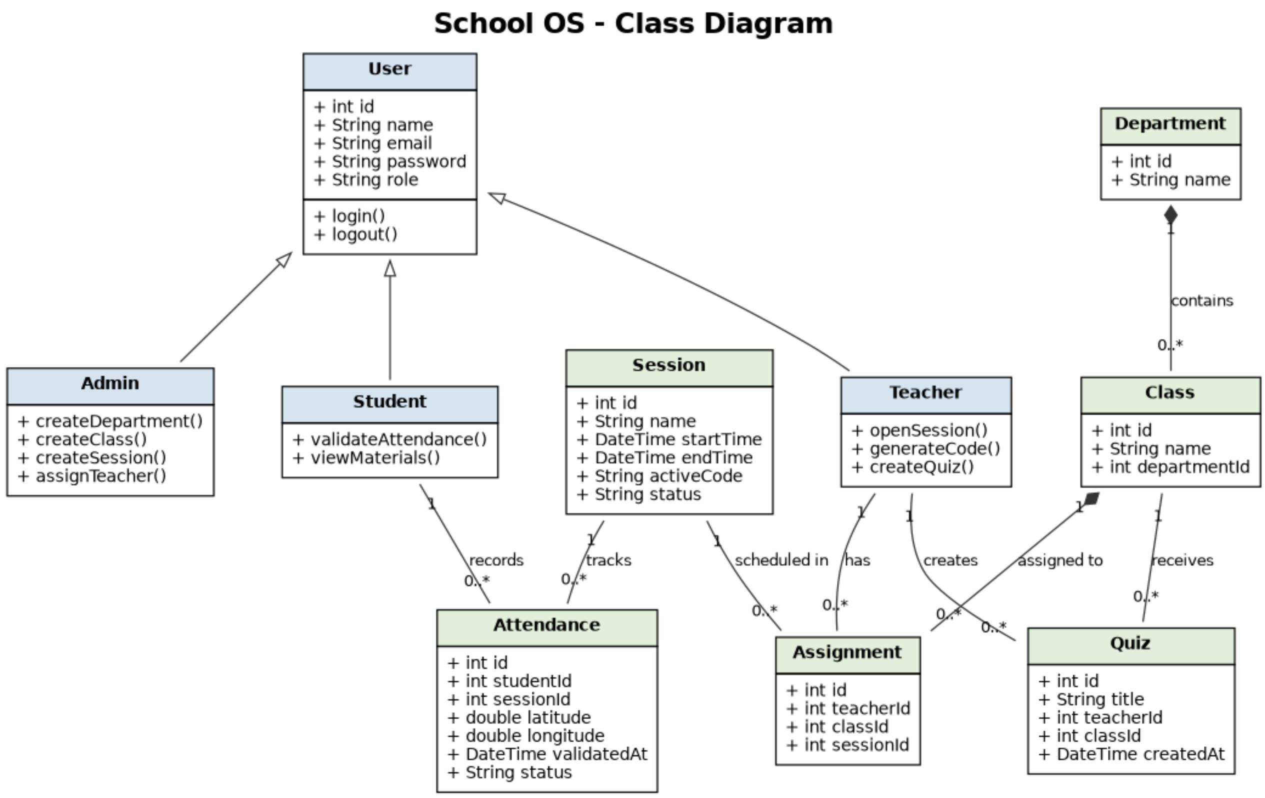

Figure 2.15: Class Diagram "student attendance tracking"

Relational Model

User (iduser, name, email, password, role)

Admin (iduser#)

Teacher (iduser#)

Student (iduser#)

Department (id, name)

Class (id, name, departmentId#)

Session (id, name, startTime, endTime, activeCode, status)

Assignment (id, teacherId#, classId#, sessionId#)

Attendance (id, studentId#, sessionId#, latitude, longitude, validatedAt, status)

Quiz (id, title, teacherId#, classId#, createdAt)

Legend: underlined = primary key ; attribute# = foreign key.
Inheritance: Admin.id, Teacher.id, Student.id are each at once the primary key and a foreign key referencing User(id).
Foreign keys: Class.departmentId → Department(id) ; Assignment.(teacherId → Teacher, classId → Class, sessionId → Session) ; Attendance.(studentId → Student, sessionId → Session) ; Quiz.(teacherId → Teacher, classId → Class).

2.5 Conclusion

At the end of this chapter, we have completed the analysis and design phase of the attendance tracking system by presenting the different UML diagrams and the overall architecture of the platform. We also explained the main functionalities of the system, including attendance validation, QR code generation, user management, session scheduling, and dashboard monitoring. In addition, the database structure and the relationships between the different entities were introduced to ensure efficient data management. In the following chapter, we will focus on the implementation phase, explaining how the proposed design is transformed into a fully functional attendance tracking system.

Chapter 3

Implementation

3.1 Introduction

In this chapter, we will design a web platform for electronic voting. We will present the tools, programming languages, and operating system used in the development process, as well as some interface examples that represent the main functions of the implemented platform.

3.2 Development Environment

In this section, we present the software tools, programming languages, development platforms, and the operating system used in the implementation of our electronic voting platform.

3.2.1 Characteristics of a Computer

Device Name	**Features**
DELL	RAM: 8GB
	CPU: Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz (2.11 GHz)
	Hard Disk: 300GB HDD
	Operating System: Windows 11

Table 3.1: Used hardwares

3.2.2 Software Tools

1. **HTML:**

HTML is the standard markup language used to create the structure of web pages. It defines the basic layout and elements such as headings, paragraphs, images, and input forms [22].

Figure 3.1: HTML

2. **CSS:**

CSS is used to style HTML elements, making the interface more appealing and organized. It defines visual aspects such as layout, colors, fonts, and spacing [23].

Figure 3.2: CSS

3. **JavaScript:**

JavaScript is a client-side programming language used to add interactivity and dynamic behavior to web pages. It enhances user experience by responding to events and managing logic on the client side [24].

Figure 3.3: JavaScript

4. **PHP:**

PHP (Hypertext PreProcessor) is a programming language whose main application is the management of dynamic websites. It can create the content of HTML pages according to different parameters. The capabilities of PHP extend beyond web pages to include image manipulation, PDF creation, and database connectivity [25].

Figure 3.4: PHP

5. **XAMPP:**

XAMPP (X (cross) Apache MariaDB Perl PHP) is a set of software to easily set up a web server and a File Transfer Protocol (FTP) server. This is a distribution of free software with good flexibility of use, known for its simple and fast installation [26].

Figure 3.5: XAMPP

6. **Visual Studio Code:**

Visual Studio Code (VS Code) is a source code editor created by Microsoft, compatible with Windows, Linux, macOS, and web browsers. It offers features such as debugging support, syntax highlighting, intelligent code completion, code snippets, code refactoring, and integrated version control using Git [27].

Figure 3.6: VS Code

7. **Grok :**

xAI Grok is an artificial intelligence system developed by xAI, a company founded by Elon Musk. It is used to help users answer questions, search for information, write texts, and provide fast and accurate responses. Grok is known for its ability to understand natural language and interact with people in a simple and easy way. It is also connected to the X platform, which allows it to access recent posts and updated information. Grok is considered one of the modern technologies in the field of artificial intelligence that aims to make human life easier and improve communication and learning methods [28].

Figure 3.7: Grok

3.3 Presentation of the Main Interfaces

This section presents the main interfaces of the application, organized by user role. We begin with the shared authentication page, then present the administrator's dashboard and its related interfaces, followed by the teacher's dashboard and its interfaces, and finally the student's interface.

3.3.1 Authentication

The login page is the common entry point for all users. Each user signs in with an email address and a password; the side panel summarizes the platform's main features (GPS-verified attendance, an AI study assistant, course/TD/TP materials, and quizzes and schedules). A link also allows new users to create an account.

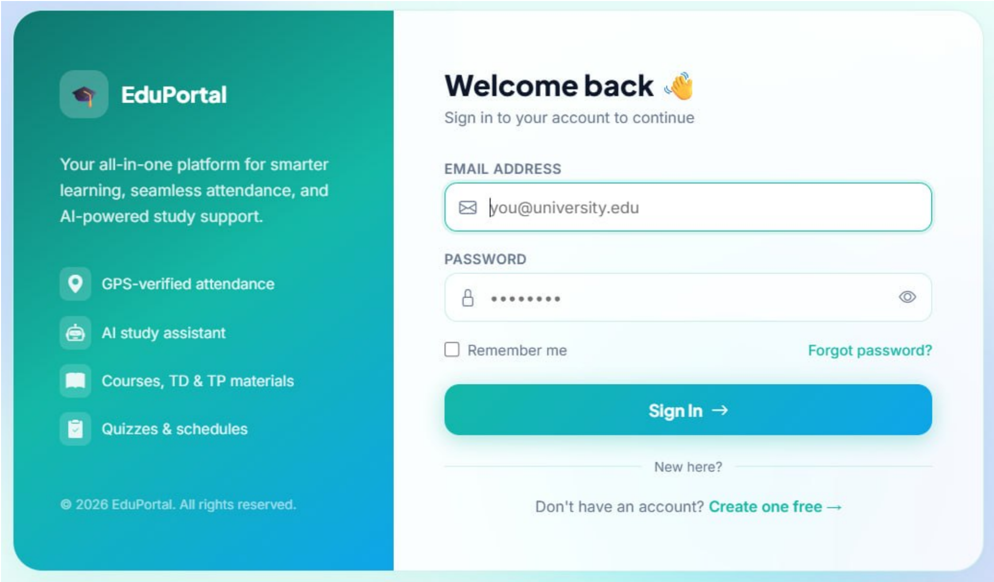

Figure 3.8: Authentication (login) interface.

3.3.2 Administrator Interfaces

Dashboard

The administrator dashboard provides a global overview of the system. A set of cards displays the key figures (total users, departments, classes, and sessions), while a ``Quick Stats" panel breaks the user population down into students, teachers, and administrators. From the left-hand menu, the administrator can reach every management function of the platform.

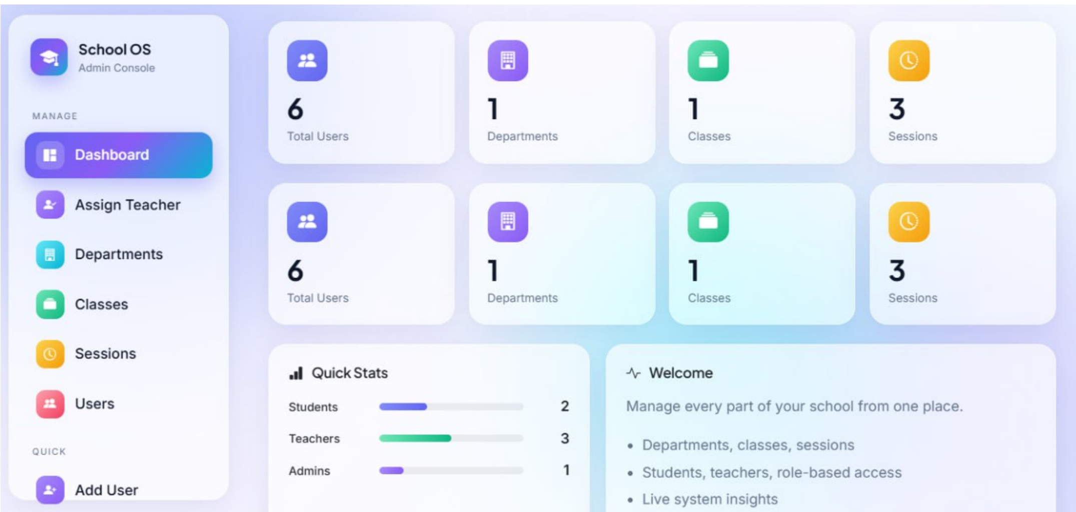

Figure 3.9: Administrator dashboard.

Department Management

This interface allows the administrator to manage the academic departments. A simple form on the left adds a new department by its name, while the table on the right lists the departments already created.

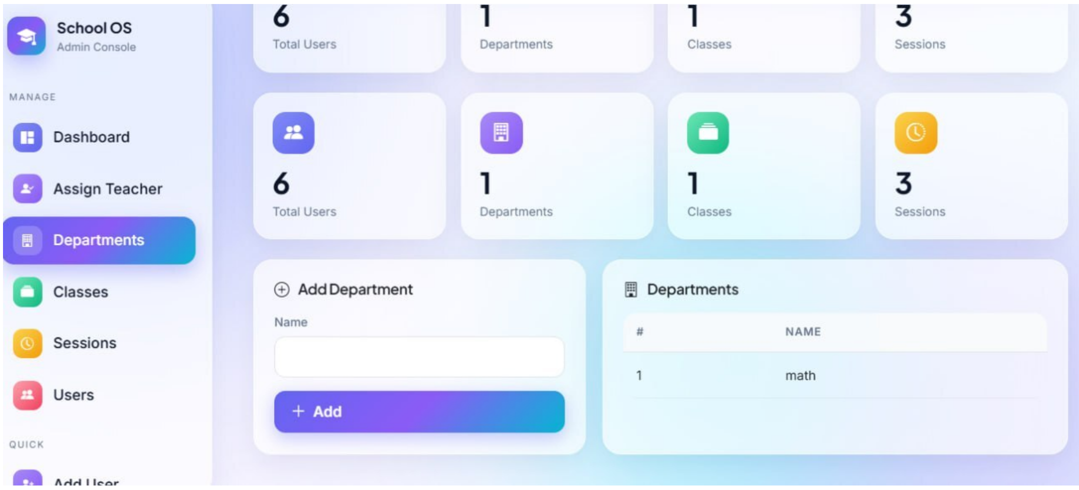

Figure 3.10: Department management.

Class Management

Here the administrator creates the classes (groups). The form specifies the class name and the department to which it belongs, and the table on the right displays each class together with its department.

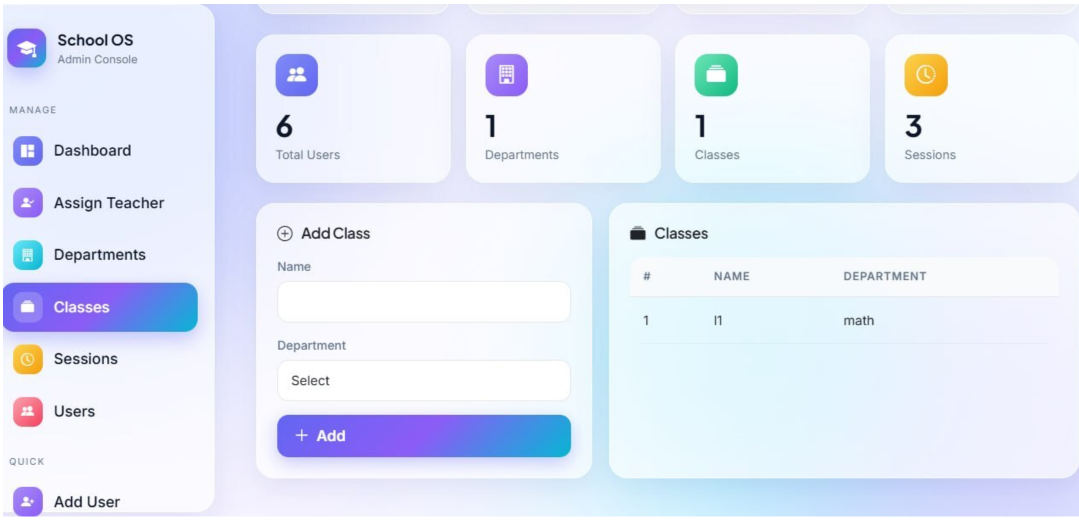

Figure 3.11: Class management.

Session Management

This interface is used to define the time sessions. For each session, the administrator enters a name and the start and end date-times; the table on the right lists the existing sessions with their start and end times.

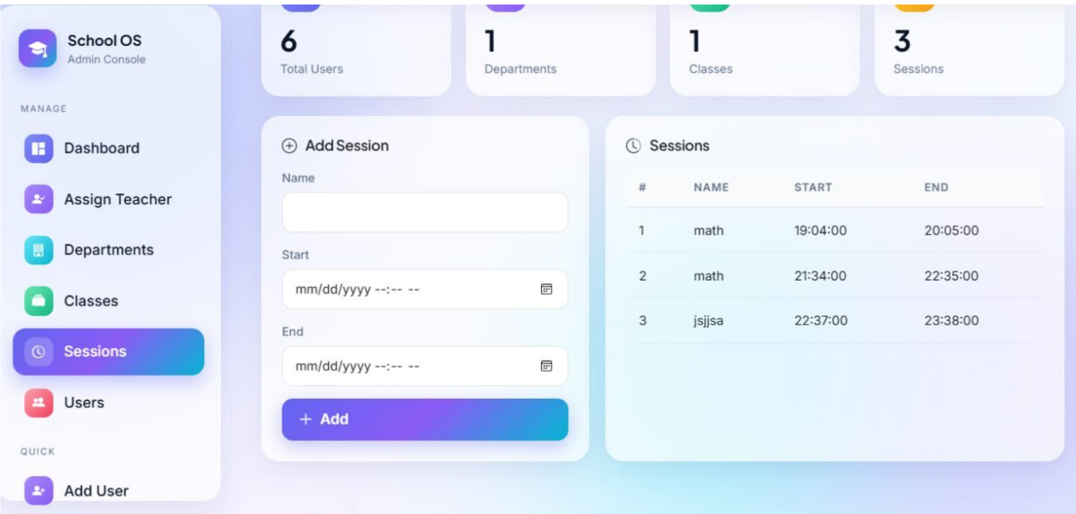

Figure 3.12: Session management.

Assigning Teachers

This interface links a teacher to a class and a session. The administrator selects the teacher, the class, and the session, then confirms the assignment; the ``Current Assignments" table summarizes the teacher--class--session links already defined.

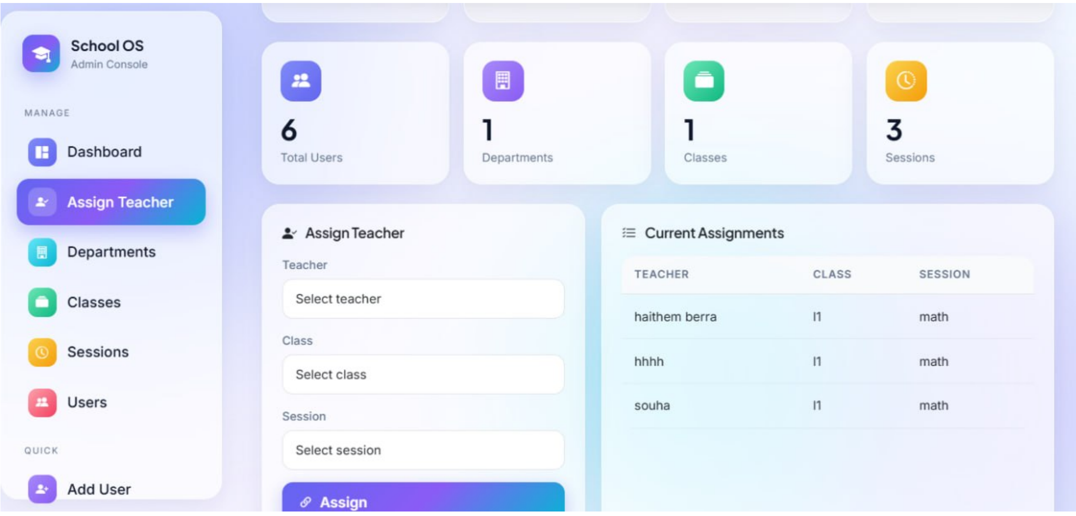

Figure 3.13: Assigning teachers to classes and sessions.

User Management

This interface manages the system's users. The central table lists the registered users, while a side form allows a new student or teacher to be added by specifying the role, full name, email, password, identification code, and phone number.

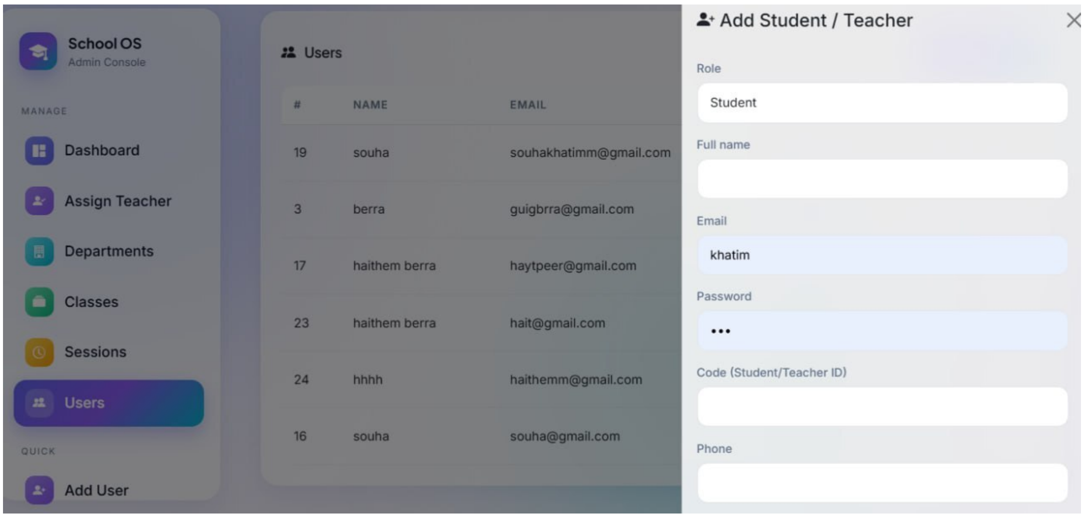

Figure 3.14: User management (adding a student or a teacher).

3.3.3 Teacher Interfaces

Dashboard

The teacher dashboard summarizes the teacher's activity through several indicator cards (number of sessions, students, validated attendances, classes, and quizzes). A set of tabs (Overview, Sessions, Quizzes, AI History) organizes the teacher's tools, and the list of the teacher's sessions allows a session to be opened directly.

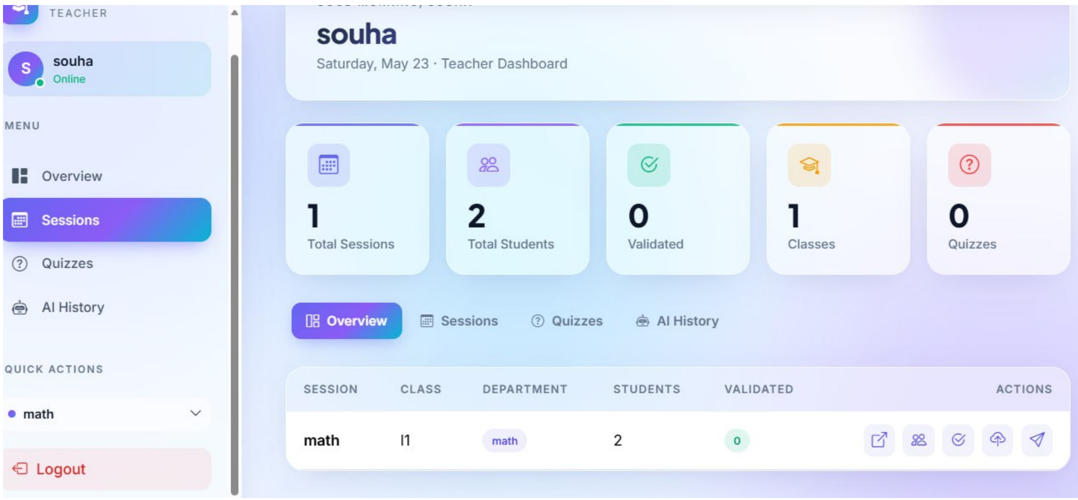

Figure 3.15: Teacher dashboard.

Opening a Session and Generating the Attendance Code

When a session is opened, the teacher sees the number of students, the number of validated attendances, and the attendance rate. The ``Attendance Code" panel lets the teacher generate a code that the students use to validate their presence, while the list of validated students is updated in real time.

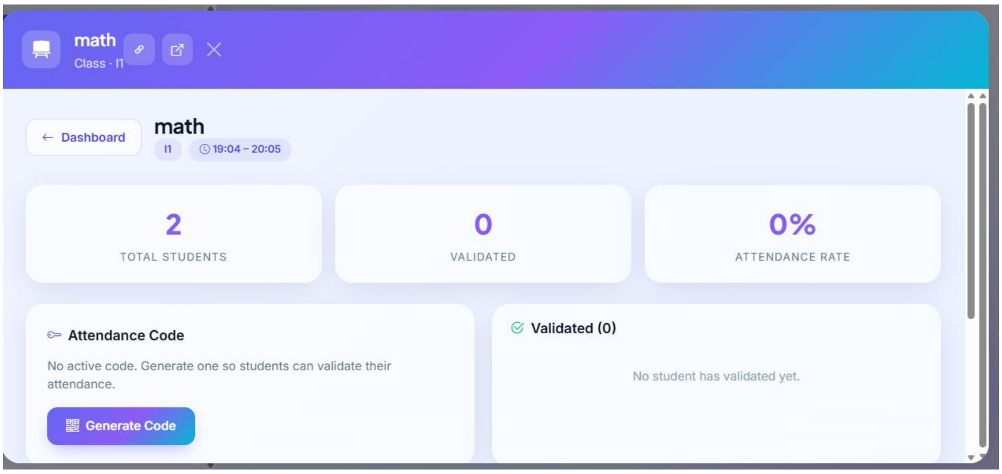

Figure 3.16: Opening a session and generating the attendance code.

Quiz Management

This interface allows the teacher to add and manage the quizzes intended for the class. The ``Add Quiz'' button uploads a new quiz, and the list displays the quizzes already created.

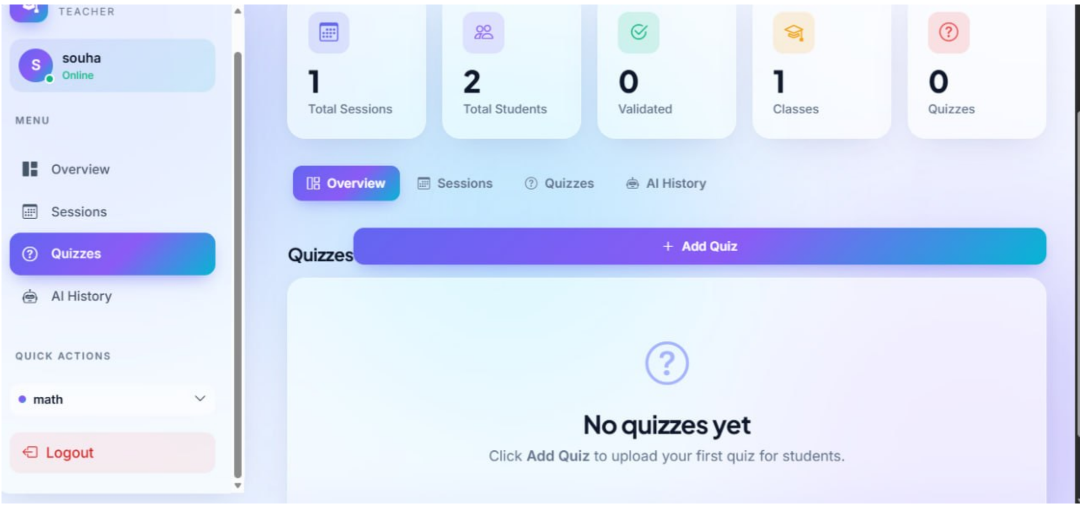

Figure 3.17: Quiz management.

AI History

This interface lets the teacher monitor the questions that students have asked the AI study assistant, providing insight into how the students use this feature.

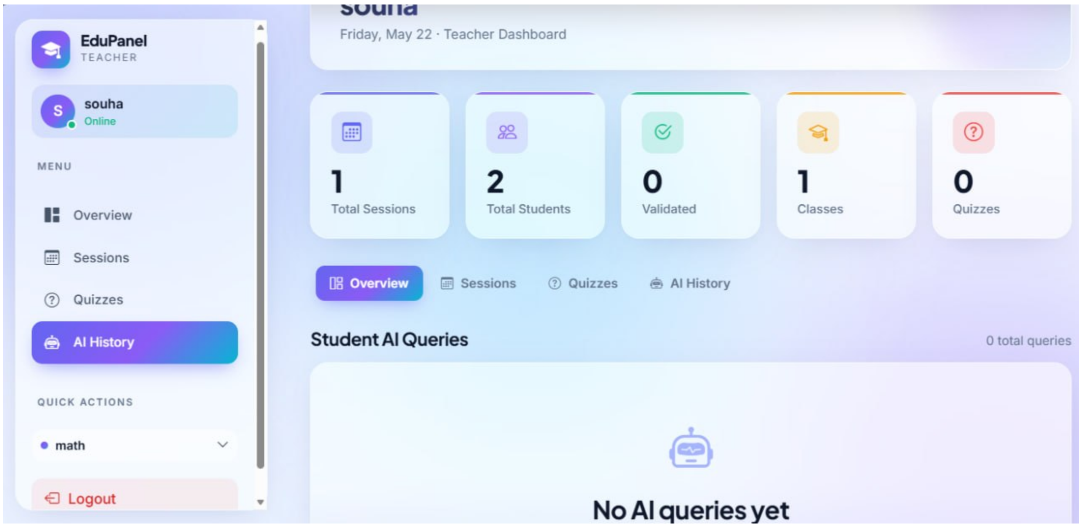

Figure 3.18: History of the students' AI assistant queries.

3.3.4 Student Interface

The student dashboard is rather long, so it is presented here in two parts: the upper part (overview and profile) and the lower part (recent materials and tools).

Dashboard -- Overview and Profile

The upper part of the student dashboard greets the student by name and shows the current date and class group. Four indicator cards summarize the student's situation: the number of available materials, the number of sessions attended out of the total, the attendance rate, and the number of quizzes assigned. Below them, a profile card recalls the student's identity (code, group, department, and phone number). A highlighted banner informs the student when the teacher has an active session and invites them to confirm their presence through the ``Validate Now" button.

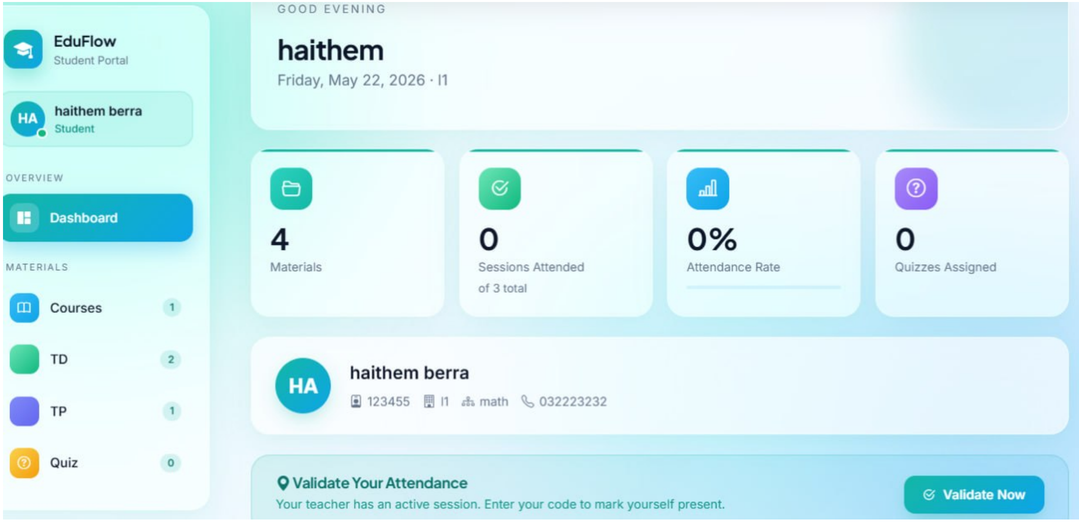

Figure 3.19: Student dashboard (upper part): overview indicators and profile.

Dashboard -- Recent Materials and Tools

The lower part of the same dashboard lists the most recent materials shared by the teachers. Each card indicates the type of resource (Course, TD, or TP), its title, its author, and its date, and offers a ``Download" button. The left-hand menu groups the student's tools: access to the courses, the TD, the TP, and the quizzes; the attendance section (schedule, history, and attendance validation); and the tools section, which includes the AI study assistant and the sign-out option.

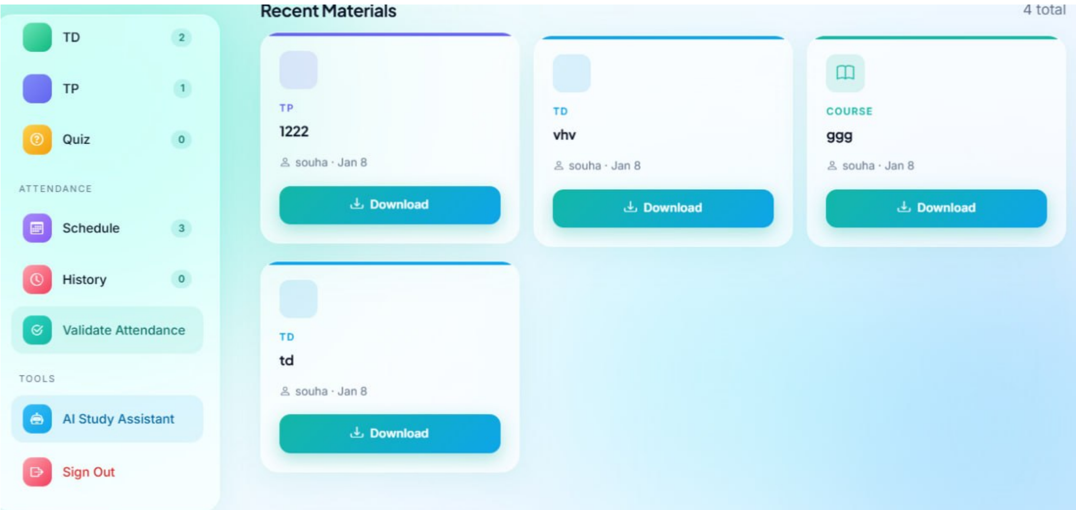

Figure 3.20: Student dashboard (lower part): recent materials and side menu.

Attendance Validation

To validate attendance, the student opens the dedicated window, enters the code provided by the teacher, and allows the application to capture the current position by GPS (latitude, longitude, and accuracy) on a map. Attendance is confirmed only when an active session exists and both the code and the location are valid.

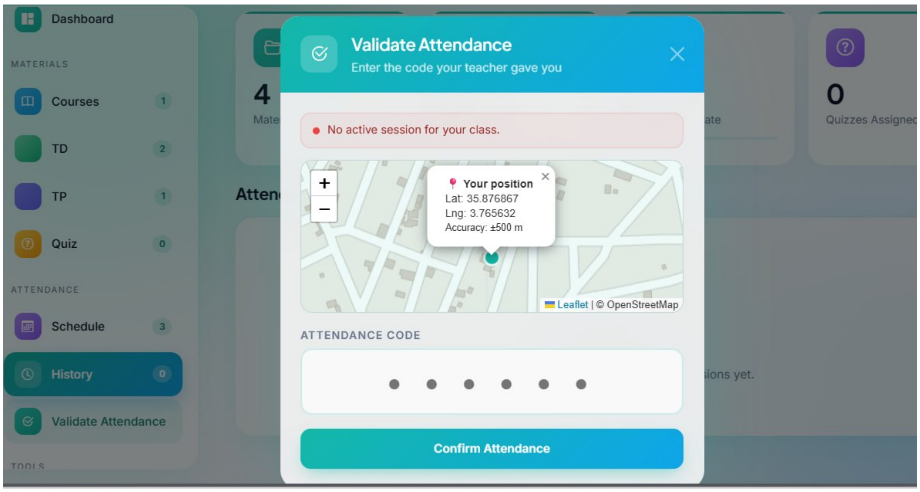

Figure 3.21: Student interface: attendance validation with GPS verification.

Conclusion

In conclusion, the development of this web application for attendance tracking and course distribution has been a highly enriching experience that allowed us to consolidate the theoretical knowledge acquired during our studies through concrete, practical application. The project served as a tangible example of how academic learning, in software engineering, database design, and web development, can be effectively translated into a tool that addresses a real institutional need, deepening our appreciation for the interplay between theory and practice.

Throughout the course of the project, we navigated a range of technical, organizational, and time-related challenges, from modelling the system and designing a coherent data model that unifies attendance and course distribution, to reconciling the requirements of three different categories of users and the regulations governing attendance in Algerian universities. These obstacles strengthened our abilities in analytical thinking, problem-solving, and decision-making, while also enhancing our capacity to work efficiently as a team and under pressure. The project further provided us with hands-on experience using modern web technologies and development tools, broadening our understanding of the field and exposing us to various professional practices.

Moreover, the project cultivated a strong sense of personal and academic responsibility. It encouraged us to engage in independent research and critical inquiry, both of which are vital components of academic excellence. We also gained a deeper appreciation for the importance of clear documentation, structured planning, and strict adherence to ethical and legal guidelines, particularly when dealing with students' personal data and attendance records, whose confidentiality and integrity must be carefully protected.

Overall, we take pride in what has been accomplished. The challenges encountered and the knowledge gained throughout this process have not only enhanced our technical competence but also contributed meaningfully to our intellectual and personal development.

Perspectives

The future development of the attendance tracking and course distribution platform is guided by a vision focused on enhancing reliability, intelligence, and accessibility. Several advancements are anticipated across both the technological and operational dimensions:

- **Integration of Artificial Intelligence (AI):** AI could be employed to analyze attendance patterns, detect irregularities in real time (such as suspected proxy attendance), and provide earlywarning indicators that identify at-risk students before they reach the threshold of exclusion, allowing timely pedagogical intervention.

- **Automatic Timetable Generation:** Building on the course-distribution module, optimization algorithms (such as genetic algorithms and other metaheuristics) could be integrated to generate conflict-free timetables automatically, while leaving the administrator in control of soft preferences.
- **Native Mobile Application:** A dedicated mobile application for Android and iOS would make attendance marking, material access, and notifications even more convenient for students and teachers, complementing the web interface.
- **Additional Attendance Methods:** Beyond QR codes and GPS verification, biometric options (fingerprint or facial recognition) could be offered as alternatives, giving institutions flexibility according to their budget and privacy obligations.
- **Two-Factor Authentication (2FA):** Strengthening user authentication through dynamic codes or biometric verification would further reduce the risk of unauthorized access and identity fraud, especially for administrator and teacher accounts.
- **Multilingual Interface Support:** To ensure inclusivity, the platform could offer multilingual support, initially in Arabic, French, and English, catering to the linguistic context of Algerian universities.
- **Integration with Official University Systems:** Real-time integration with the national higher-education platform (PROGRES) and the institution's student information system would synchronize enrolments, groups, and modules automatically, prevent duplicate data entry, and align the exported absence and exclusion lists with official procedures.
- **Security and Data Protection:** Cybersecurity and privacy would be embedded at every layer of the design, from encrypting sensitive personal data to regular security testing and continuous monitoring, ensuring the system remains trustworthy as it scales to larger numbers of users.

Through these enhancements, the platform aspires not only to digitize attendance and course distribution but to evolve into a comprehensive, intelligent academic-management tool that is more reliable, inclusive, and aligned with the needs of the modern university.

References

[1] Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, ``Portail des plateformes numériques / progres,'' https://progres.mesrs.dz/, 2025, accessed: 2026-05-30.

[2] T. Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. HarperCollins, 2000.

[3] L. Richardson and M. Amundsen, *RESTful Web APIs*. O'Reilly Media, 2013.

[4] S. Alhassan, ``A web-based attendance management system using a microcontroller,'' Master's thesis, Savonia University of Applied Sciences, 2024.

[5] M. Widenius and D. Axmark, *MySQL Reference Manual*. O'Reilly Media, 2002.

[6] Z. Kotevski, N. Blazheska-Tabakovska, and T. Dimovski, ``Web-based solution for student attendance tracking: Development and evaluation,'' *Applied Internet and Information Technologies*, 2020.

[7] J. D. Ani, K. K. Flores, C. S. K. Otadoy, J. P. Rodas *et al.*, ``Classroom attendance monitoring system using radio frequency identification with web application,'' *Research Briefs on Information and Communication Technology Excellence*, 2020.

[8] N. Nordin and Y. Z. Hong, ``A real-time web-based facial recognition attendance system,'' *Multidisciplinary Applied Research and Innovation*, 2022.

[9] L. Newman-Ford, K. Fitzgibbon, S. Lloyd, and S. Thomas, ``A large-scale investigation into the relationship between attendance and attainment: a study using an innovative, electronic attendance monitoring system,'' *Studies in Higher Education*, vol. 33, no. 6, pp. 699--717, 2008.

[10] E. Bowen, T. Price, S. Lloyd, and S. Thomas, ``Improving the quantity and quality of attendance data to enhance student retention,'' *Journal of Further and Higher Education*, vol. 29, no. 4, pp. 375--385, 2005.

[11] R. Meier, *Professional Android 4 Application Development*. John Wiley & Sons, 2012.

[12] R. Lewis, ``A survey of metaheuristic-based techniques for university timetabling problems,'' *OR Spectrum*, vol. 30, no. 1, pp. 167--190, 2008.

[13] T. Müller, H. Rudová, and Z. Müllerová, ``Real-world university course timetabling at the international timetabling competition 2019,'' *Journal of Scheduling*, vol. 28, no. 2, pp. 247--267, 2025.

[14] People's Democratic Republic of Algeria, ``Law no. 99-05 of 4 april 1999 on higher education,'' 1999, official Gazette of Algeria.

[15] Joint Declaration of the European Ministers of Education, ``Joint declaration of the european ministers of education convened in bologna on 19 june 1999,'' https://ehea.info/Upload/document/ministerial_declarations/1999_Bologna_Declaration_English_553028.pdf, 1999.

[16] People's Democratic Republic of Algeria, ``Executive decree no. 04-371 of 21 november 2004 introducing the lmd system,'' 2004, official Gazette of Algeria.

[17] Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, ``Ministerial decision no. 1165 of 4 october 2025 on registration, organisation of training, and assessment in the licence and master tracks,'' 2025, cited from the institutional regulation used in the thesis; public official URL not located.

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

[19] S. Brown, *The C4 Model for Software Architecture*. Leanpub, 2018.

[20] Object Management Group (OMG), ``OMG Unified Modeling Language (OMG UML) Version 2.5.1,'' 2017.

[21] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed. Addison-Wesley, 2004.

[22] MDN Web Docs Contributors, ``Html reference,'' https://developer.mozilla.org/en-US/docs/Web/HTML/Reference, 2025, last modified 2025-07-09; accessed 2026-05-30.

[23] E. A. Meyer, *CSS: The Definitive Guide*. O'Reilly Media, 2007.

[24] D. Flanagan, *JavaScript: The Definitive Guide*, 6th ed. O'Reilly Media, 2011.

[25] K. Tatroe, P. MacIntyre, and R. Lerdorf, *Programming PHP*, 3rd ed. O'Reilly Media, 2011.

[26] Apache Friends, *XAMPP: Apache + MariaDB + PHP + Perl*, 2024.

[27] Microsoft Corporation, *Visual Studio Code Documentation*, 2024.

[28] xAI Corporation, ``Grok: Frontier ai models for everything you build,'' https://x.ai/grok, 2026, accessed: 2026-05-30.