



ESPRIM

ClassCheck Absence Manager

Academic Attendance Management System

Development Team:

Mokni Hamdi

Oussama Fekih Hassen

 <https://github.com/Mk-1000/ClassCheck>

 hamdimokni@gmail.com

 hssanoussama@gmail.com

January 13, 2025

Executive Summary

ClassCheck Absence Manager represents a state-of-the-art .NET Core application designed to revolutionize attendance management in educational institutions. This comprehensive system leverages cutting-edge technologies to deliver:

- Authentication for secure user login and management
- Attendance tracking and monitoring
- Real-time analytics and reporting
- Role-based access control
- Seamless integration capabilities
- Mobile-responsive interface

Built on .NET Core 6.0 and utilizing modern software architecture principles, the system ensures scalability, maintainability, and security while providing an intuitive user experience for all stakeholders.

Contents

Executive Summary	1
1 Introduction	5
1.1 Project Overview	5
1.2 Problem Statement	5
1.3 Project Objectives	5
2 System Analysis and Design	6
2.1 Requirements Analysis	6
2.1.1 Functional Requirements	6
2.1.2 Non-Functional Requirements	6
3 Technical Architecture	7
3.1 Technology Stack	7
3.2 Class Diagram	7
3.3 Core Entities	9
3.3.1 Entity Overview	9
3.3.2 Detailed Entity Models	9
3.3.3 Database Context Implementation	10
3.3.4 Entity Relationship Model	11
4 User Interface	12
4.1 Authentication and Authorization	12
4.1.1 Identity Implementation	12
4.1.2 Role-Based Authorization	12
4.2 Home Page	13
4.3 Dashboard	14
4.4 Absence Management Page	15
5 Future Enhancements	16
5.1 Planned Features	16
5.2 Technology Roadmap	16
6 Conclusion	17
6.1 Project Achievements	17
6.2 Lessons Learned	17
References	18

List of Figures

3.1	Class Diagram for ClassCheck Absence Manager	8
3.2	Entity Relationship Diagram Generated by DBeaver	11
4.1	Login Page	12
4.2	Home Page for Admin	13
4.3	Dashboard Overview (Admin Role)	14
4.4	Dashboard Overview (Student Role)	14
4.5	Absence Management Page	15

List of Tables

3.1	Technology Stack Specification	7
5.1	Development Roadmap	16

Introduction

1.1 Project Overview

ClassCheck Absence Manager is a state-of-the-art attendance management system designed to modernize and streamline educational administrative processes. This sophisticated solution leverages modern .NET technologies to provide:

- Secure authentication for user login and management
- Real-time attendance tracking and monitoring
- Comprehensive reporting and analytics
- Role-based access control
- Mobile-responsive user interface

1.2 Problem Statement

Traditional attendance management systems face numerous challenges that impact educational institutions' efficiency and effectiveness. Key issues include:

- ⚠ Time-consuming paper-based processes
- ⚠ Limited accessibility to historical attendance data
- ⚠ Inefficient reporting mechanisms
- ⚠ Lack of real-time insights and analytics
- ⚠ Difficulty in maintaining attendance records across multiple courses

1.3 Project Objectives

The ClassCheck system aims to address these challenges through the following objectives:

1. **Accessibility:** Provide anywhere, anytime access to attendance data
2. **Accuracy:** Ensure data integrity and reduce human error
3. **Analytics:** Deliver comprehensive reporting and analytical capabilities
4. **Security:** Implement robust role-based access control with secure authentication

System Analysis and Design

2.1 Requirements Analysis

2.1.1 Functional Requirements

- ✔ **User Management**
 - User authentication
 - Role-based access control
 - Profile management
- ✔ **Attendance Management**
 - Record and track student attendance
 - Manage course schedules
 - Handle multiple class sections
- ✔ **Reporting System**
 - List of attendance reports
 - Create statistical analyses

2.1.2 Non-Functional Requirements

- **Performance:** Response time \leq 2 seconds
- **Scalability:** Support for 10,000+ concurrent users
- **Reliability:** 99.9
- **Security:** Data encryption and secure authentication
- **Usability:** Intuitive interface with minimal training required

Technical Architecture

3.1 Technology Stack

primaryblue		
Backend Framework	.NET Core	6.0
Database	MySQL	8.0
ORM	Entity Framework Core	6.0
Frontend	Razor Pages	6.0
CSS Framework	Bootstrap	5.0
JavaScript	jQuery	3.6
Authentication	ASP.NET Identity	6.0

Table 3.1: Technology Stack Specification

3.2 Class Diagram

- **Departement**: A department that organizes classes and employs teachers.
- **Groupe**: A group that organizes multiple classes.
- **Classe**: A class containing students and attendance records.
- **Matiere (Subject)**: A subject associated with attendance records for a class.
- **Grade**: A teacher's rank or position within a department.
- **User**: A system user (teacher or student) with credentials for authentication.
- **Enseignant (Teacher)**: A teacher assigned to classes and responsible for attendance.
- **Etudiant (Student)**: A student enrolled in a class, whose attendance is tracked.
- **Seance (Session)**: A specific lecture session for a subject.
- **FicheAbsence (Attendance Record)**: A class attendance record for a particular day.
- **FicheAbsenceSeance (Session Attendance Record)**: A record for attendance during a specific session.
- **LigneFicheAbsence (Attendance Line)**: A record of whether a student was present or absent for a session.

The ClassCheck Absence Manager system is based on a relational data model that efficiently manages attendance across various entities. The class diagram below illustrates the core classes and their relationships within the system.

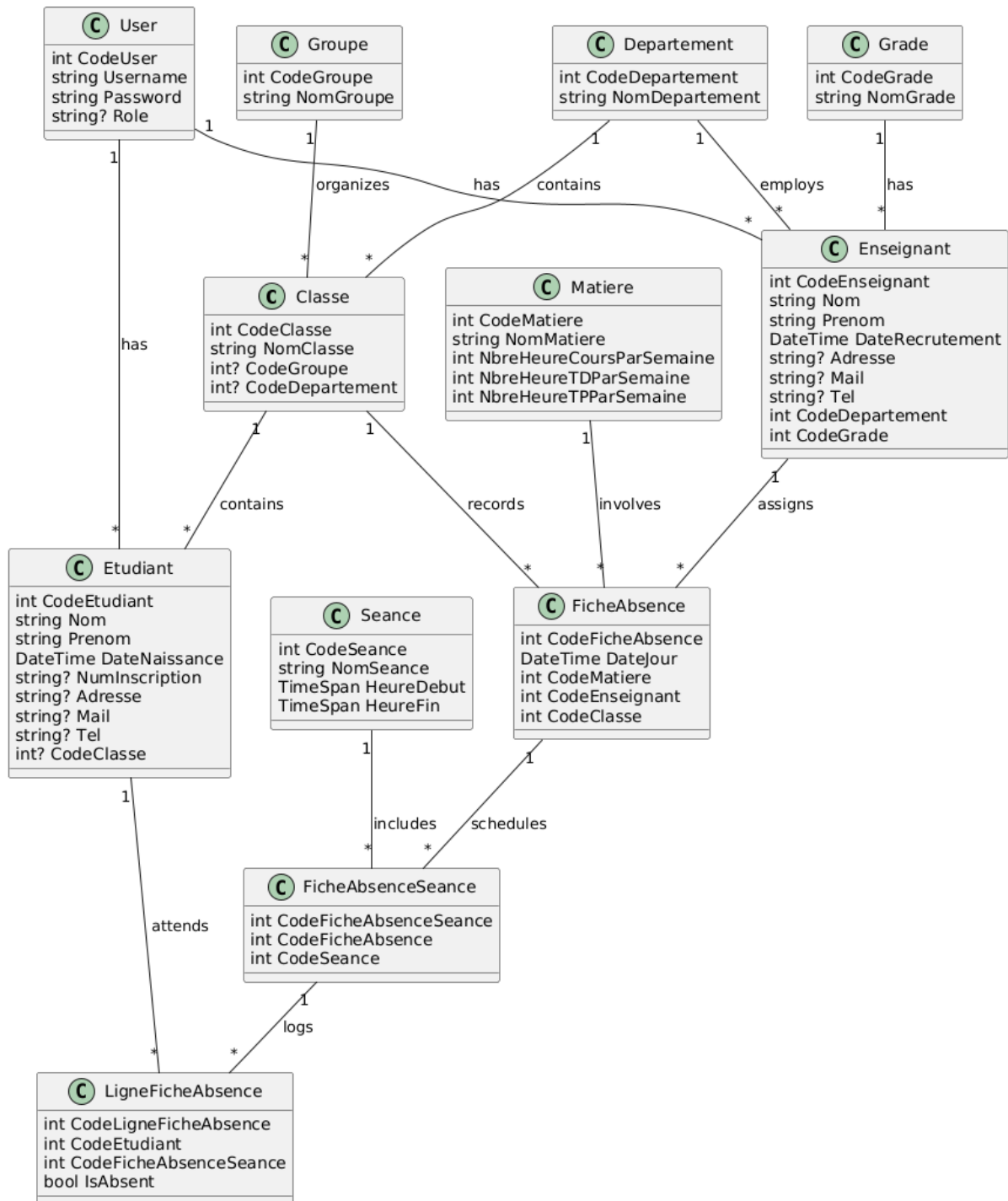


Figure 3.1: Class Diagram for ClassCheck Absence Manager

3.3 Core Entities

This section outlines the core entities of the system, detailing their attributes, relationships, and roles within the overall architecture. These entities enable robust data management and operational efficiency.

3.3.1 Entity Overview

The system's data models are grouped into the following categories:

- **Student Management:** Entities such as 'Etudiant' and 'Classe' manage student information and class assignments.
- **Attendance Tracking:** Entities like 'FicheAbsence', 'FicheAbsenceSeance', and 'LigneFicheAbsence' track and manage attendance records.
- **Faculty and Department Management:** Entities such as 'Enseignant', 'Departement', and 'Grade' manage faculty information and departmental structure.
- **Course and Session Management:** Entities like 'Matiere' and 'Seance' handle course details and session timings.
- **User Management:** The 'User' entity manages user authentication and roles.

3.3.2 Detailed Entity Models

Student Management

Student Entity ('Etudiant'): The 'Etudiant' entity captures personal and academic information about students. It is linked to 'Classe' for class assignments and to 'LigneFicheAbsence' for attendance records.

```

1 public class Etudiant
2 {
3     [Key]
4     public int CodeEtudiant { get; set; }
5     public string Nom { get; set; }
6     public string Prenom { get; set; }
7     public DateTime DateNaissance { get; set; }
8     public string? NumInscription { get; set; }
9     public string? Adresse { get; set; }
10    public string? Mail { get; set; }
11    public string? Tel { get; set; }
12    [ForeignKey("Classe")]
13    public int? CodeClasse { get; set; }
14    // Navigation properties
15    public Classe? Classe { get; set; }
16    public ICollection<LigneFicheAbsence> LignesFicheAbsence { get; set
17    ; } = new List<LigneFicheAbsence>();

```

Listing 3.1: Student Entity Model

3.3.3 Database Context Implementation

The ‘MyDbContext’ class defines the database schema, including relationships and constraints. Below is the implementation:

```
1 public class MyDbContext : DbContext
2 {
3     public MyDbContext(DbContextOptions<MyDbContext> options) : base(
4         options) { }
5
6     // DbSet properties
7     public DbSet<Classe> Classes { get; set; }
8     public DbSet<Seance> Seances { get; set; }
9     ...
10
11    // OnModelCreating configuration
12    protected override void OnModelCreating(ModelBuilder modelBuilder)
13    {
14        // Primary Key configurations
15        modelBuilder.Entity<Classe>().HasKey(c => c.CodeClasse);
16        modelBuilder.Entity<Groupe>().HasKey(g => g.CodeGroupe);
17        ...
18
19        // Composite Key configurations
20        modelBuilder.Entity<FicheAbsenceSeance>()
21            .HasKey(fas => fas.CodeFicheAbsenceSeance);
22        ...
23
24        // Relationships and Foreign Key configurations
25        modelBuilder.Entity<Classe>()
26            .HasOne(c => c.Groupe)
27            .WithMany(g => g.Classes)
28            .HasForeignKey(c => c.CodeGroupe)
29            .OnDelete(DeleteBehavior.SetNull);
30        ...
31    }
```

Listing 3.2: Database Context Implementation

3.3.4 Entity Relationship Model

Figure 3.2 illustrates the relationships among the core entities in the system, captured using DBeaver’s database visualization tool. This diagram represents the MySQL database schema that underpins our application.

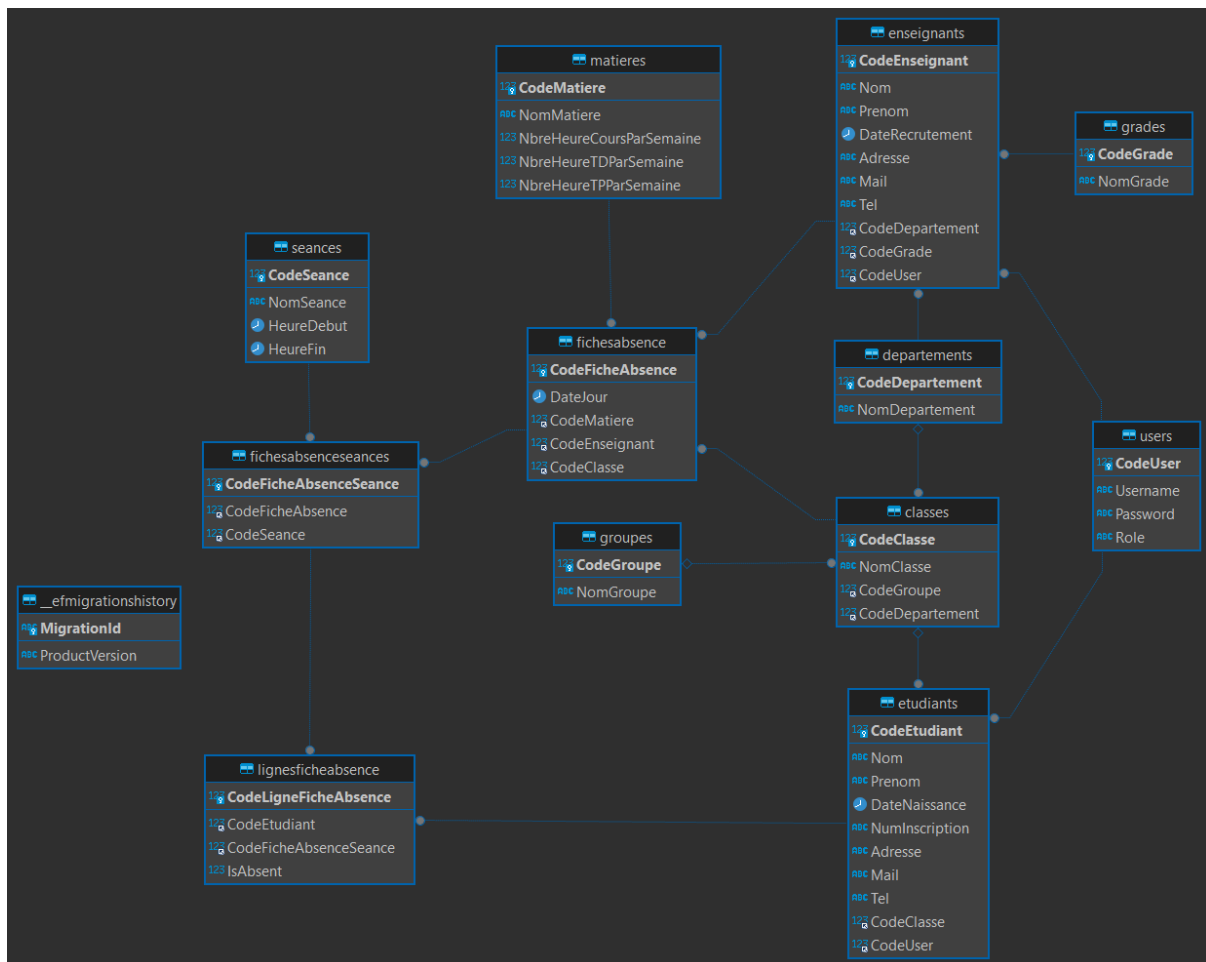


Figure 3.2: Entity Relationship Diagram Generated by DBeaver

User Interface

4.1 Authentication and Authorization

4.1.1 Identity Implementation

The ClassCheck system implements ASP.NET Core Identity for robust authentication and authorization. This framework provides:

- User account management
- Role-based authorization

4.1.2 Role-Based Authorization

The system implements role-based access control with the following primary roles:

- **Administrator:** Full system access and management capabilities
- **Teacher:** Access to attendance management and reporting
- **Student:** Limited access to view personal attendance records

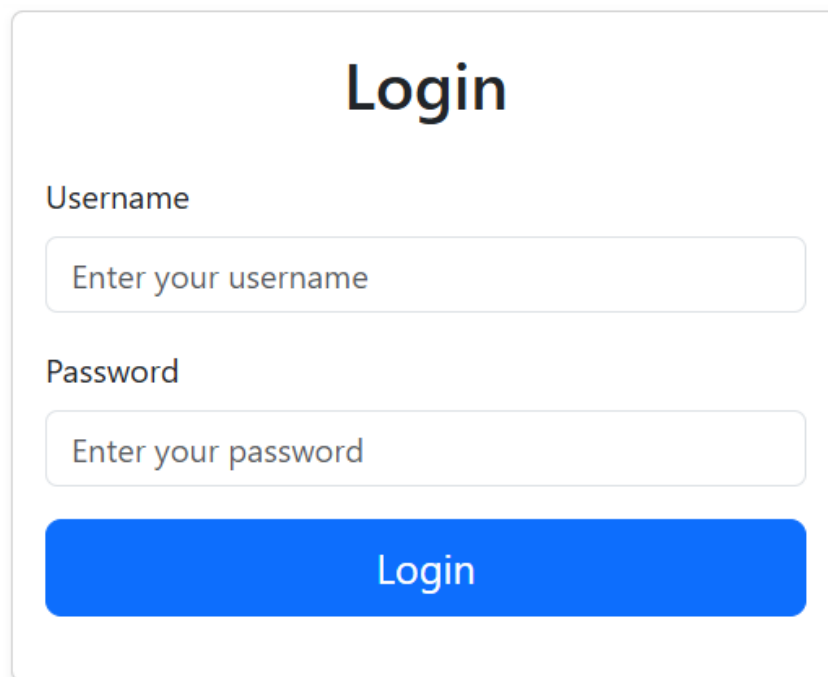
A login form with a light gray background and rounded corners. At the top, the word "Login" is centered in a large, bold, black font. Below it, the label "Username" is positioned to the left of a white input box with a light gray border. Inside the box, the placeholder text "Enter your username" is displayed in a light gray font. Below the username box, the label "Password" is positioned to the left of another white input box with a light gray border. Inside this box, the placeholder text "Enter your password" is displayed in a light gray font. At the bottom of the form, there is a solid blue button with the word "Login" centered in white text.

Figure 4.1: Login Page

4.2 Home Page

The home page serves as the central navigation hub for the system. It features a grid layout with interactive cards representing different management modules, such as **Etudiants**, **Classes**, **Groupes**, **Enseignants**, and more. Each card includes:

- **Title:** The name of the module (e.g., "Etudiant Management").
- **Icon:** A contextual icon for visual clarity.
- **Description:** A brief overview of the module's functionality.
- **Action Button:** A link to the respective module's management page.

Key Features:

- Clean and intuitive design for effortless navigation.
- Responsive layout ensuring usability on various devices.
- The system provides customized home pages based on user roles, ensuring that users only see relevant functions and information:
 - **Administrator:** Full access to all management modules.
 - **Teacher:** Access to attendance management.

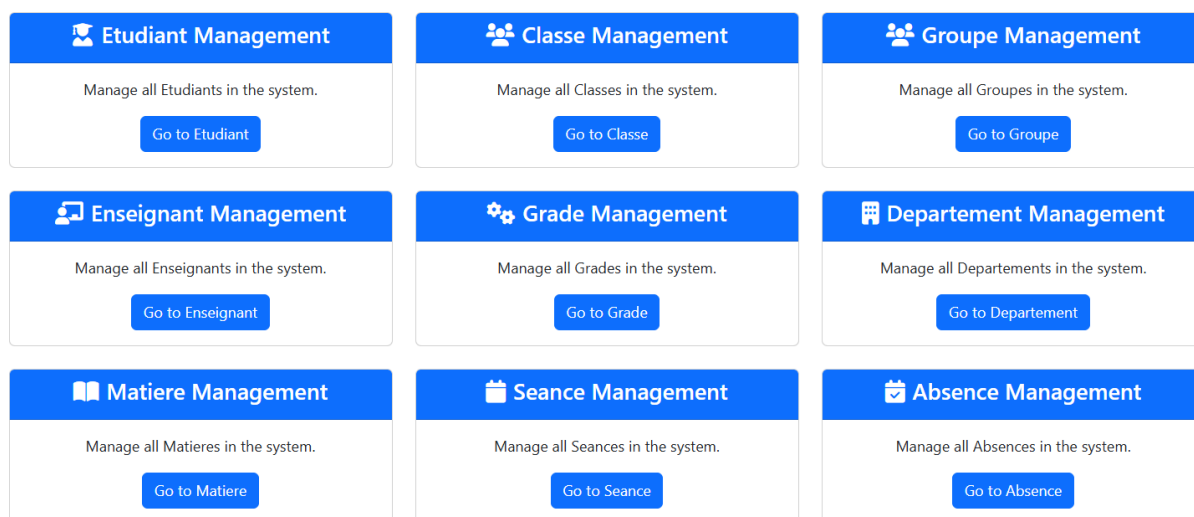


Figure 4.2: Home Page for Admin

4.3 Dashboard

The dashboard provides an overview of key statistics related to students, professors, classes, and departments, with an easy way to view recent absence records using advanced filters.

The dashboard adjusts according to the user role:

- **Admin:** Views all data, including counts of students, professors, classes, and departments.
- **Student:** Views their own absence records, filtered by name and class.
- **Professor:** Views absence records for their classes and students, with filtering options by student and date.

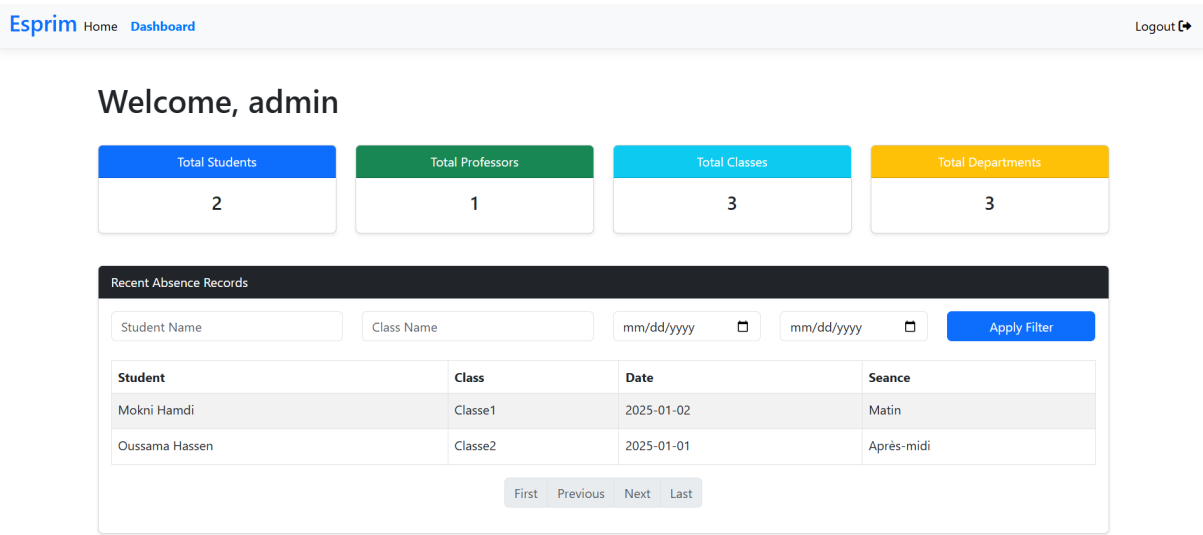


Figure 4.3: Dashboard Overview (Admin Role)

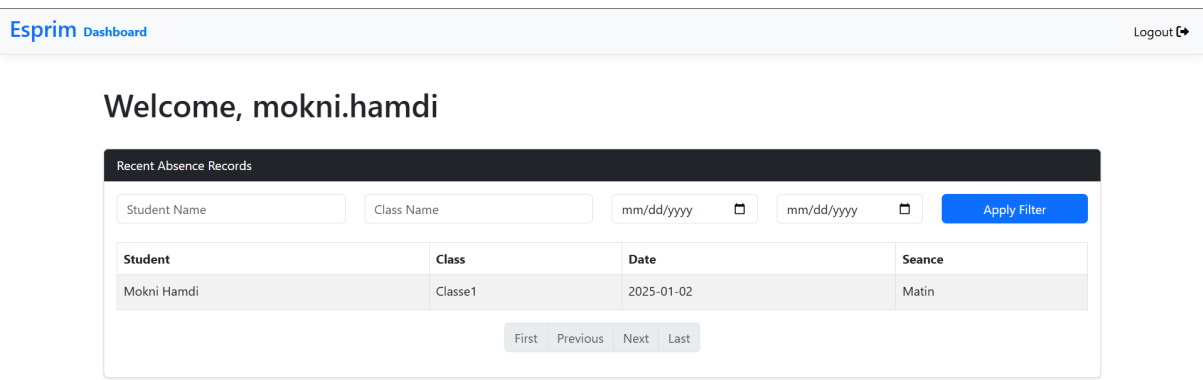


Figure 4.4: Dashboard Overview (Student Role)

4.4 Absence Management Page

The Absence Management page allows users to manage student absences effectively. It includes a table listing all attendance forms with the following columns:

- **Date:** The date of the absence.
- **Subject:** The associated subject.
- **Teacher:** The teacher responsible.
- **Class:** The class affected.
- **Sessions & Absences:** Details of sessions and absent students.
- **Actions:** Options for adding sessions, marking absences, editing, or deleting records.

Key Functionalities:

1. **Add New Attendance Record:** A modal form for creating a new absence record, capturing details like date, subject, teacher, and class.
2. **Mark Absences:** A modal for selecting absent students from a dropdown.
3. **Responsive Table:** A searchable and sortable table for quick data access.

Absence Management + Add New Absence Record					
Date	Subject	Teacher	Class	Sessions & Absences	Actions
12/12/2024	Matiere1	firas	Classe1	Matin Absent Students: • mokni • oussema	Remove Session Mark Absences
12/12/2024	Matiere1	firas	Classe1	Après-midi Absent Students: • mokni	Remove Session Mark Absences
13/12/2024	Matiere2	firas	Classe1	Après-midi Absent Students: • mokni	Remove Session Mark Absences

Figure 4.5: Absence Management Page

Future Enhancements

5.1 Planned Features

Future development plans include:

- ★ Mobile application development
- ★ Machine learning for attendance pattern analysis
- ★ Integration with learning management systems
- ★ Advanced reporting and analytics
- ★ Automated notification system
- ★ Biometric authentication integration

5.2 Technology Roadmap

primaryblue		
Phase 1	Mobile App Development	Q1 2025
Phase 2	AI/ML Integration	Q2 2025
Phase 3	Advanced Analytics	Q3 2025
Phase 4	System Integration	Q4 2025

Table 5.1: Development Roadmap

Conclusion

6.1 Project Achievements

The ClassCheck Absence Manager has successfully achieved its primary objectives:

- ✔ Improved data accuracy
- ✔ Enhanced reporting capabilities
- ✔ Streamlined administrative processes
- ✔ Increased user satisfaction

6.2 Lessons Learned

Key insights gained during development:

- Importance of thorough requirement analysis
- Benefits of modular architecture
- Need for scalable design from the start

References

Bibliography

- [1] Microsoft (2024). .NET Documentation. <https://docs.microsoft.com/en-us/dotnet/>
- [2] Microsoft (2024). Entity Framework Core Documentation. <https://docs.microsoft.com/en-us/ef/core/>
- [3] Bootstrap Team (2024). Bootstrap Documentation. <https://getbootstrap.com/docs/>
- [4] Microsoft (2024). Azure DevOps Documentation. <https://docs.microsoft.com/en-us/azure/devops/>
- [5] OWASP (2024). Web Security Guidelines. <https://owasp.org/www-project-web-security-testing-guide/>