
# CLASSROOM AUTOMATIC ATTENDANCE SYSTEM PROJECT REPORT

**Prepared by:**
**Berkay Aksoy**
**Student Number: 220706307**

**Hakkı Dökmeci**
**Student Number: 220706026**

**Course: SE 342 01 Software Validation and Testing**
**Instructor: Emre Olca**

# PROJECT CONTENTS

# Classroom Automatic Attendance System

## Requirements Analysis Document – Main Sections (EPICs)

## 1. System Requirements & Analysis

**Story:** Functional Requirements Definition

**-Subtask:** Prepare a *Functional Requirement Specification (FRS)* document describing user actions, system features, and data interactions.

**Story:** Non-Functional and System Constraints Analysis

**-Subtask:** Document *Non-Functional Requirements (NFR)* including performance, security, reliability, and scalability constraints.

## 2. Database Analysis & Setup

**Story:** Define Conceptual Database Design

**-Subtask:** Identify the main entities, attributes, and relationships required for the system before creating the ERD.

**Story:** Create Initial ERD Diagram

**-Subtask:** Create a structured list of entities, attributes, and relationship notes to be used as the ERD blueprint.

## 3. Initial System Foundations

**Story:** Database Development Setup

**-Subtask:** Create essential tables, relationships, and constraints according to the planned database model.

**Story:** Create Initial Application Screen Designs

**-Subtask:** Create basic wireframes and layouts for main screens such as login, dashboard, and attendance pages.

## 4. Course & Classroom Management

**Story:** Course Management Module

**-Subtask:** Develop CRUD (Create, Read, Update, Delete) operations for course data management..

**Story:** Classroom Assignment System

**-Subtask:** Build the classroom assignment feature; the system should automatically update timetables when changes occur.

## 5. Camera & Classroom Setup

**Story:** Camera Installation and Linking

**-Subtask:** Link each camera to its respective classroom and validate network accessibility.

**Story:** Camera Monitoring Dashboard

**-Subtask:** Build a real-time dashboard for monitoring camera status and live feed availability.

## 6. Attendance Session & Recognition Flow

**Story:** Automatic Attendance Session

**-Subtask:** Implement triggers to start and end attendance sessions based on class schedules.

**Story:** Face Recognition Attendance Flow

**-Subtask:** Integrate the face recognition module to automatically identify students and mark attendance.

## 7. Teacher Approval & Manual Adjustment

**Story:** Attendance Review Module

**-Subtask:** Create a user interface for teachers to review and confirm recognized attendance records.

**Story:** Manual Attendance Adjustment

**-Subtask:** Provide functionality for teachers to manually correct attendance errors or missing entries

## 8. Reporting & Semester Archive

**Story:** Attendance Reporting Module

**-Subtask:** Generate attendance reports by student, course, and semester.

**Story:** Semester Archive System

**-Subtask:** Develop a system to archive and export attendance data at the end of each term.

## 9. Privacy, Access, and Data Retention Policies

**Story:** Data Privacy Control

**-Subtask:** Define and implement role-based access control for teachers, administrators, and system operators.

**Story:** Data Retention & Deletion Policy

**-Subtask:** Add an automatic cleanup and secure data deletion mechanism after the retention period.

## 10. Testing & Evaluation

**Story:** System Integration Testing

**-Subtask:** Conduct integration tests between modules and document test results.

**Story:** Performance & Accuracy Evaluation

**-Subtask:** Measure face recognition accuracy, system response time, and performance metrics; compile results into a report.

## 11. Deployment & Maintenance

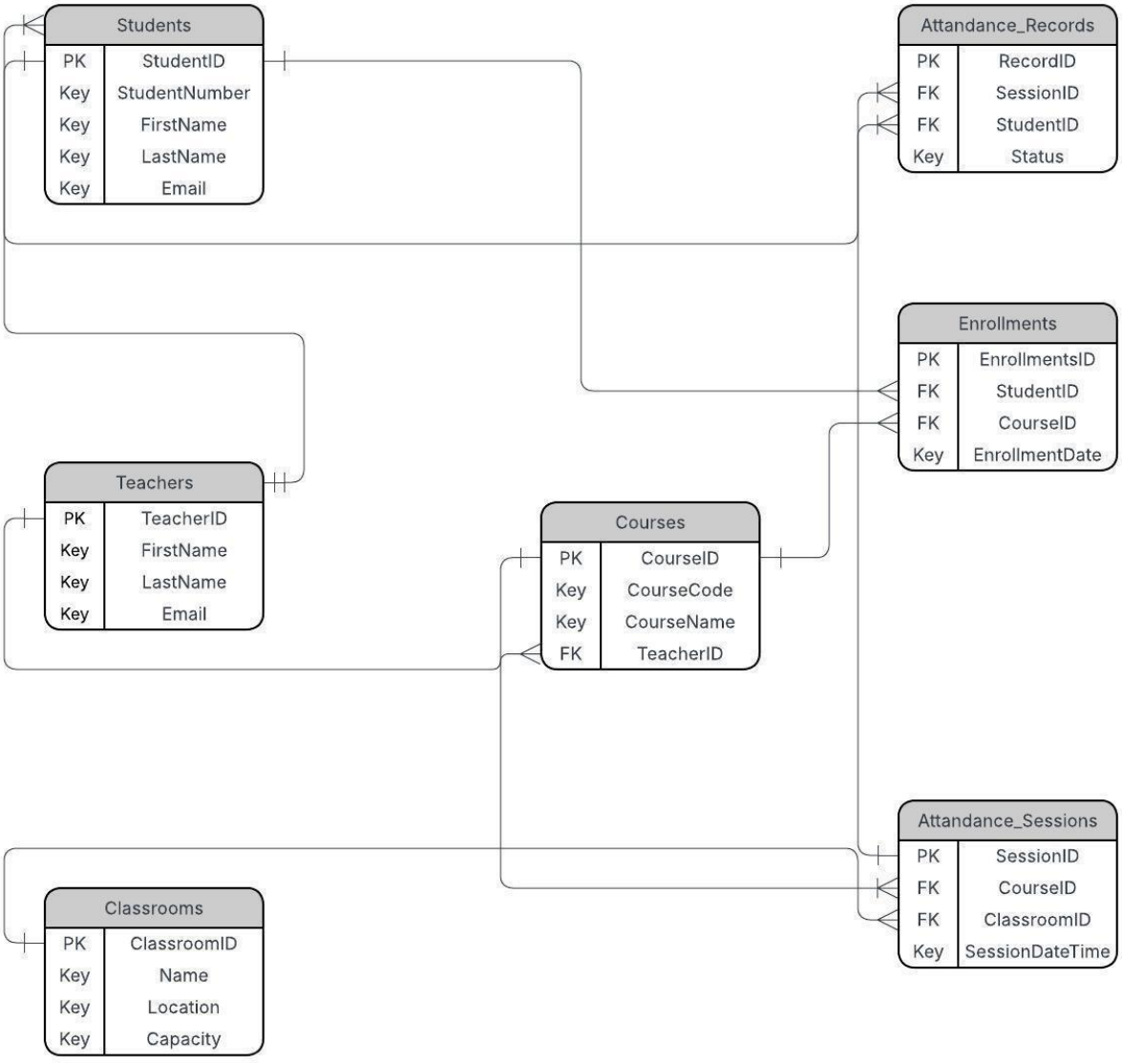
**Story:** System Deployment & Environment Setup

**-Subtask:** Deploy the system, configure environment variables, and verify installation integrity.

**Story:** System Maintenance & Update

**-Subtask:** Schedule regular maintenance, monitor system performance, and apply software updates.

# 2. Data Base Design



# 2.1 ERD Description

## 1. Students

- Stores student information such as ID, name, and email.
- Connected to **Enrollments** (courses they take).
- Linked to **Attendance_Records** (their attendance per session).

## 2. Teachers

- Holds teacher identity and contact information.
- Each teacher is responsible for multiple **Courses**.

## 3. Courses

- Defines course details, including code and name.
- Each course is taught by one **Teacher**.
- Linked to:
    - **Enrollments** (students registered)
    - **Attendance_Sessions** (scheduled attendance events)

## 4. Classrooms

- Contains physical classroom details such as name, location, and capacity.
- Each **Attendance_Session** occurs in one classroom.

## 5. Enrollments

- Resolves the many-to-many relationship between **Students** and **Courses**.
- Indicates which student is registered in which course.

## 6. Attendance_Sessions

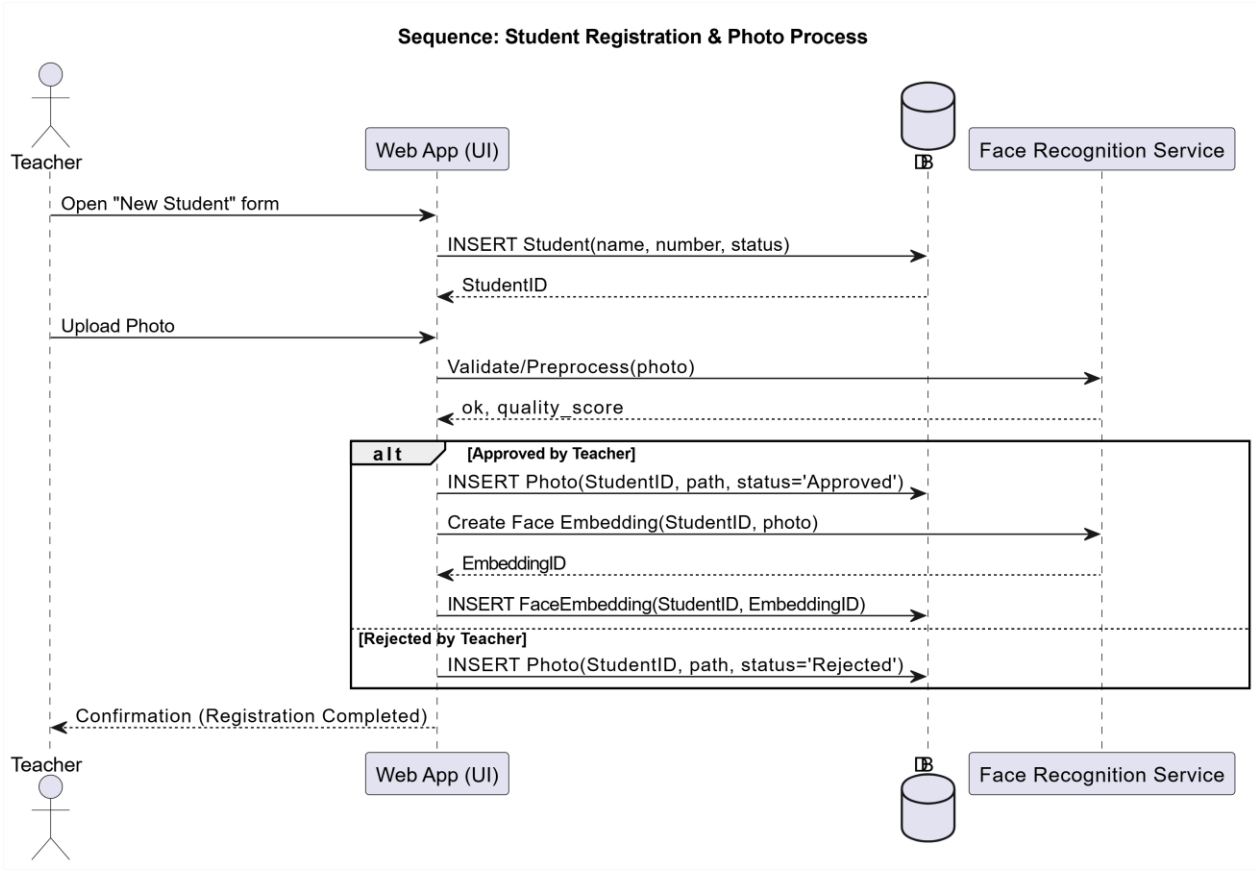
- Represents scheduled attendance sessions for a course.
- Linked to:
    - One **Course**
    - One **Classroom**
- Each session produces multiple **Attendance_Records**.

## 7. Attendance_Records

- Stores the attendance status (present/absent) for each student in each session.
- Links:
    - One **Student**
    - One **Attendance_Session**

# 3. Sequence Diagram

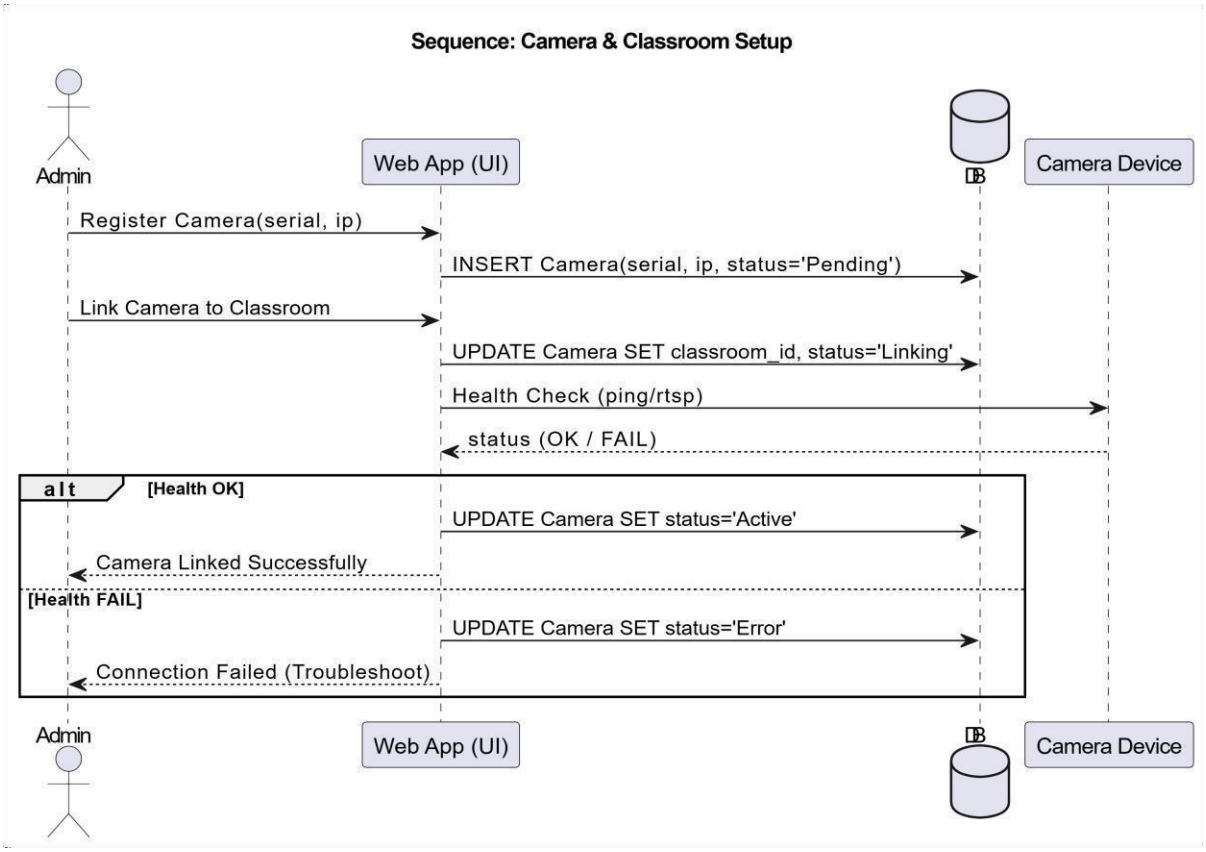
## 3.1 Sequence Diagram (Student Registration & Photo Process)




## Description:

- Thisworkflow describes how a *teacher* adds a new student, uploads a photo, and interacts with the face recognition service.
- It focuses on the **registration** stage before attendance begins.
- ***Goal:*** Prepare accurate student data for later recognition.

## 3.2 Sequence Diagram (Camera & Classroom Setup)

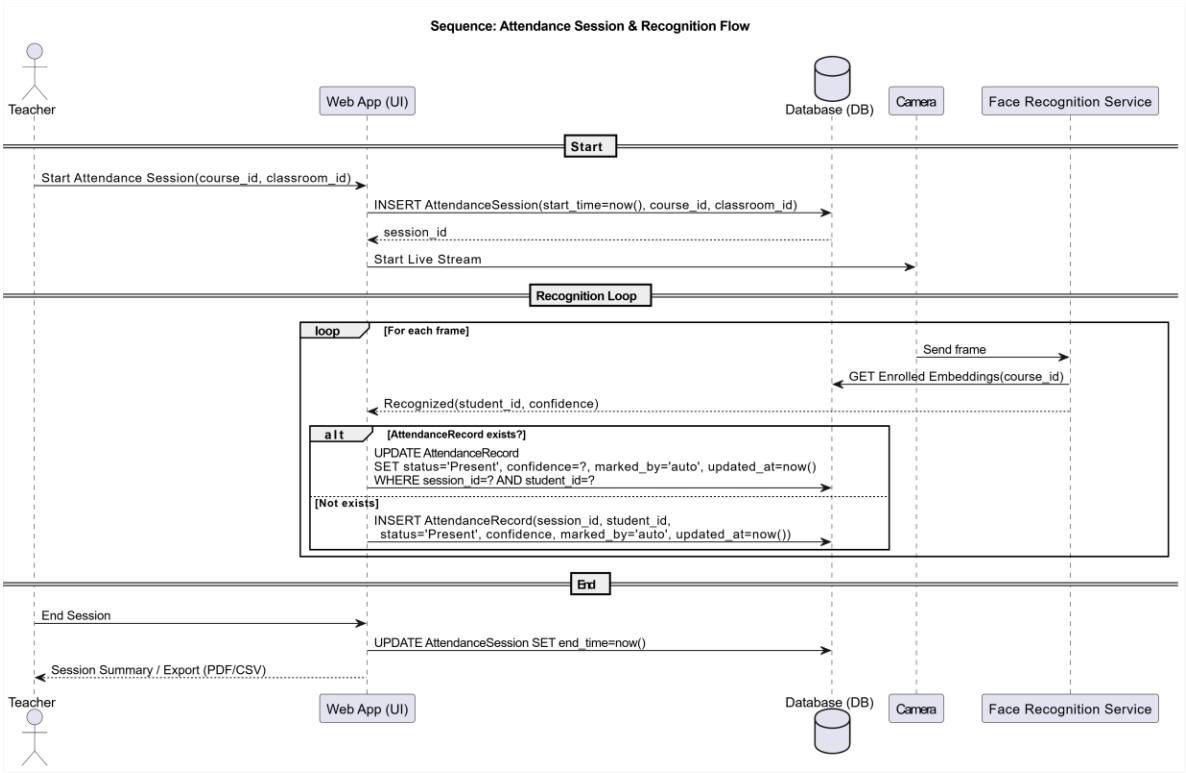



## Description:

- Thisworkflow shows how the *admin* installs and links cameras to classrooms.
- It's a completely different process — involves hardware checks and configuration, not student data.

  - ***Goal:*** Make sure the physical camera system is connected and ready to record sessions.

## 3.3 Sequence Diagram (Attandence Session & Recognition Flow)

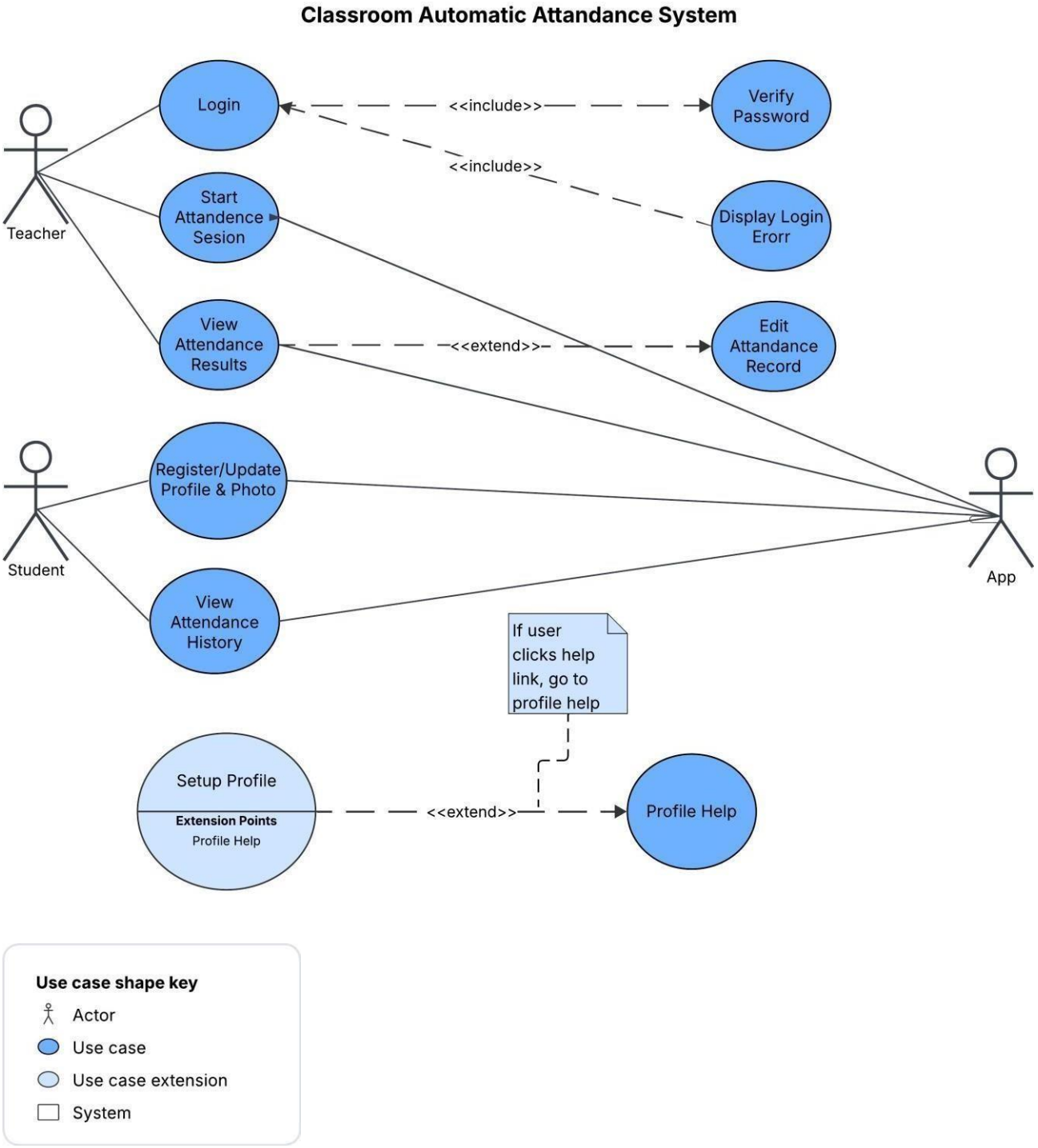



## Description:

- Thisis the main **operational phase** — where the teacher starts a session, the camera streams video, and the recognition service marks attendance.
- It dependson the previous two processes but runs as a separate daily routine.

***Goal:*** Automatically detect students and record attendance.

# 4. Use Case Diagram




## 4.1 Use-Case Diagram Description

This use-case diagram shows the main interactions in the *Classroom Automatic Attendance System*. **Teacher** and **Student** perform core actions such as logging in, managing profiles, starting attendance sessions, and viewing results or history. The **App** supports the system by handling automated tasks and optional extensions like login

validation, error display, editing attendance records, and profile help. The diagram highlights mandatory steps using **include** and optional behaviors using **extend** relationships.

### <<include>> — Role in the Diagram

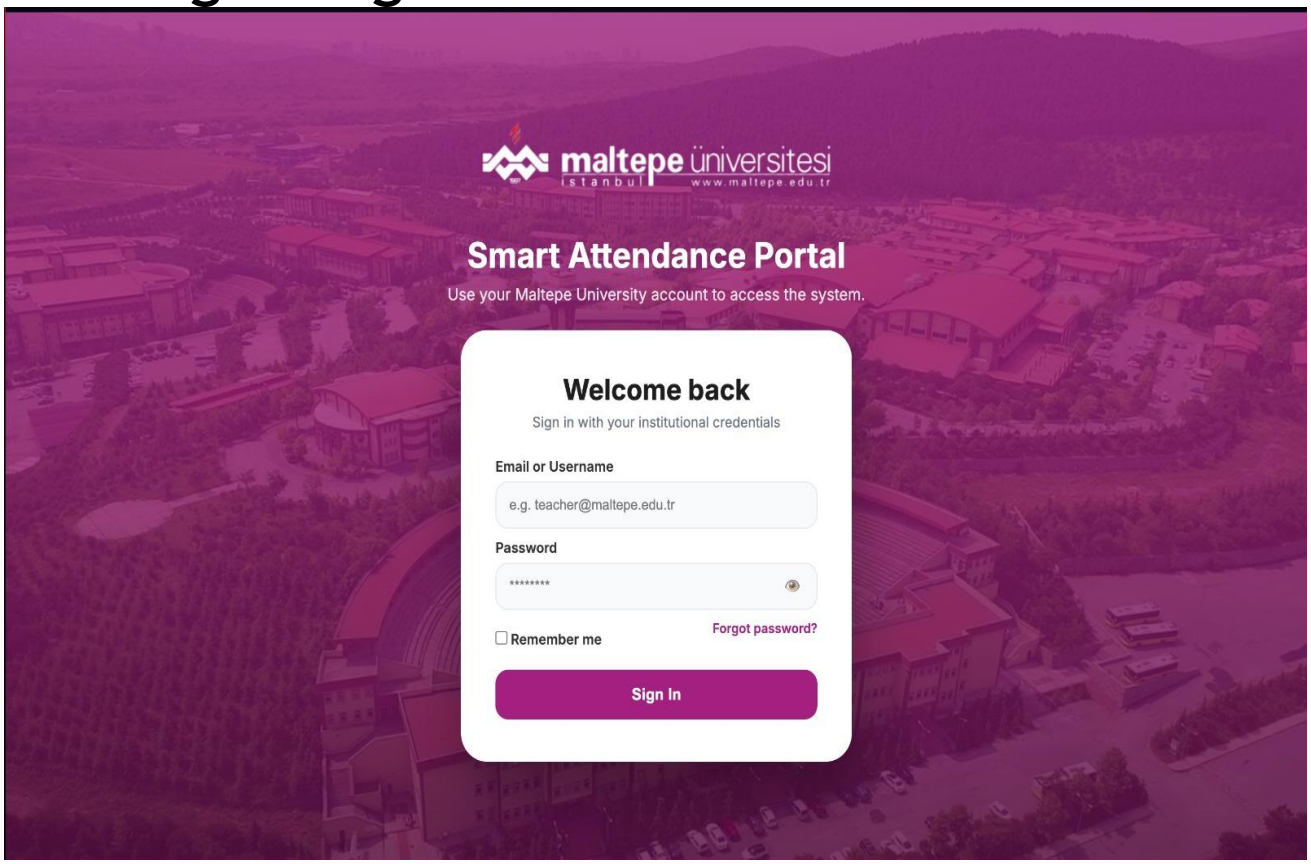
In the diagram, **<<include>>** is used to show **required sub-actions** that must occur as part of the main use case.
For example, the **Login** use case always includes **Verify Password**, and may include **Display Login Error** when needed.

### <<extend>> — Role in the Diagram

In the diagram, **<<extend>>** is used to show **optional or additional actions** that can be triggered depending on specific conditions within the main use case.
For example, **View Attendance Results** can be extended with **Edit Attendance Record**, and **Setup Profile** can be extended with **Profile Help** if needed.

# 5. Page Designs

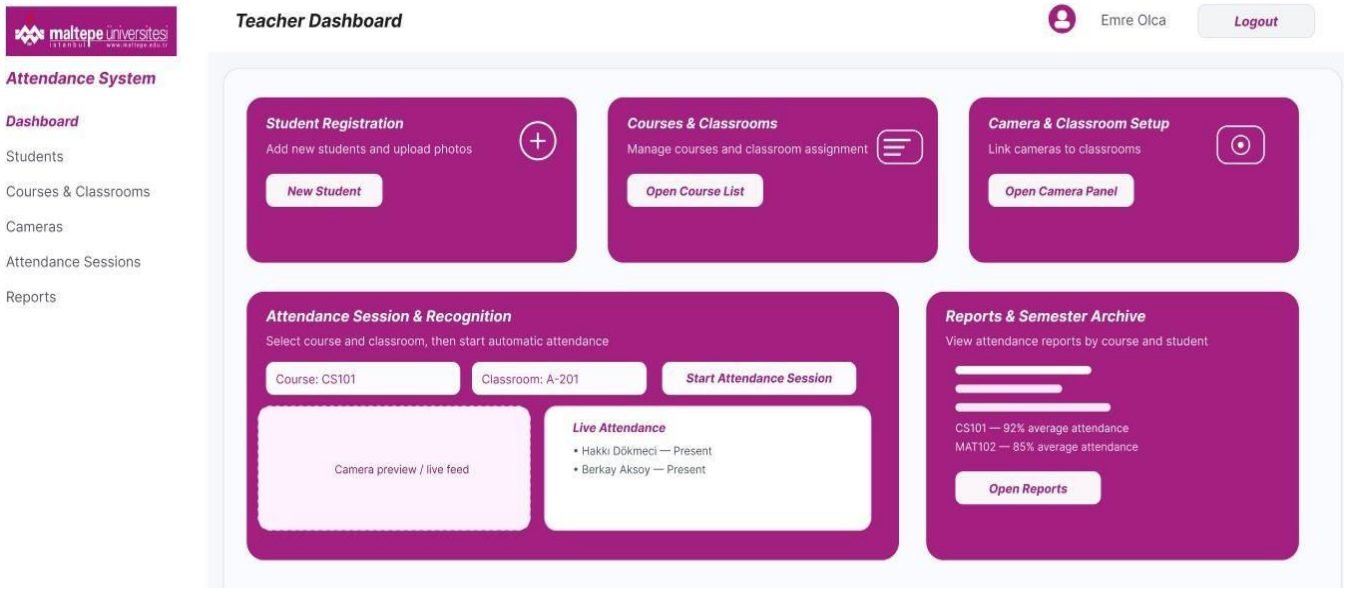
## 5.1 Login Page




Description:
The Login Page allows teachers to securely sign in with their Maltepe University

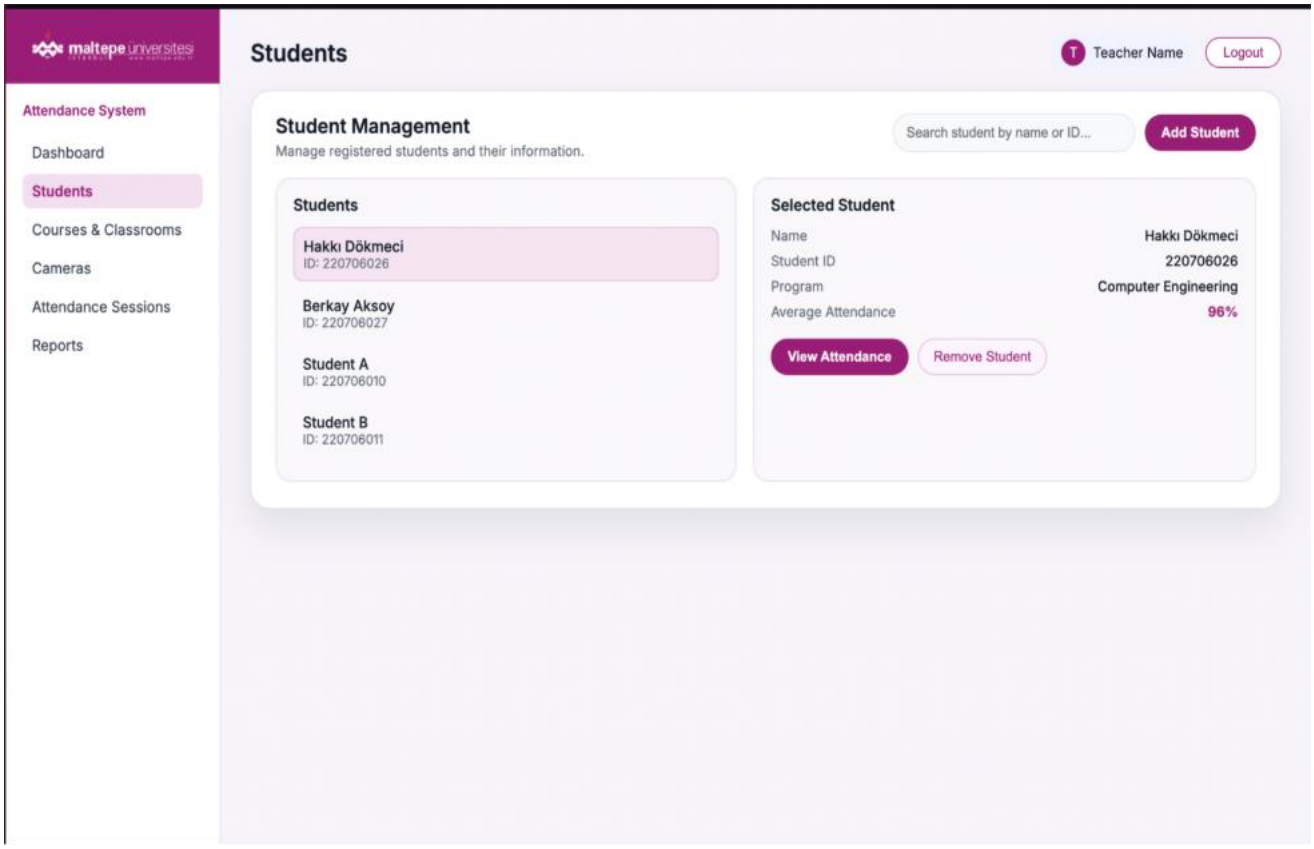
credentials. It provides a clean interface with username and password fields, a “Remember me” option, and a modern layout styled with the university’s colors.

# 5.2 Teacher Dashboard Page



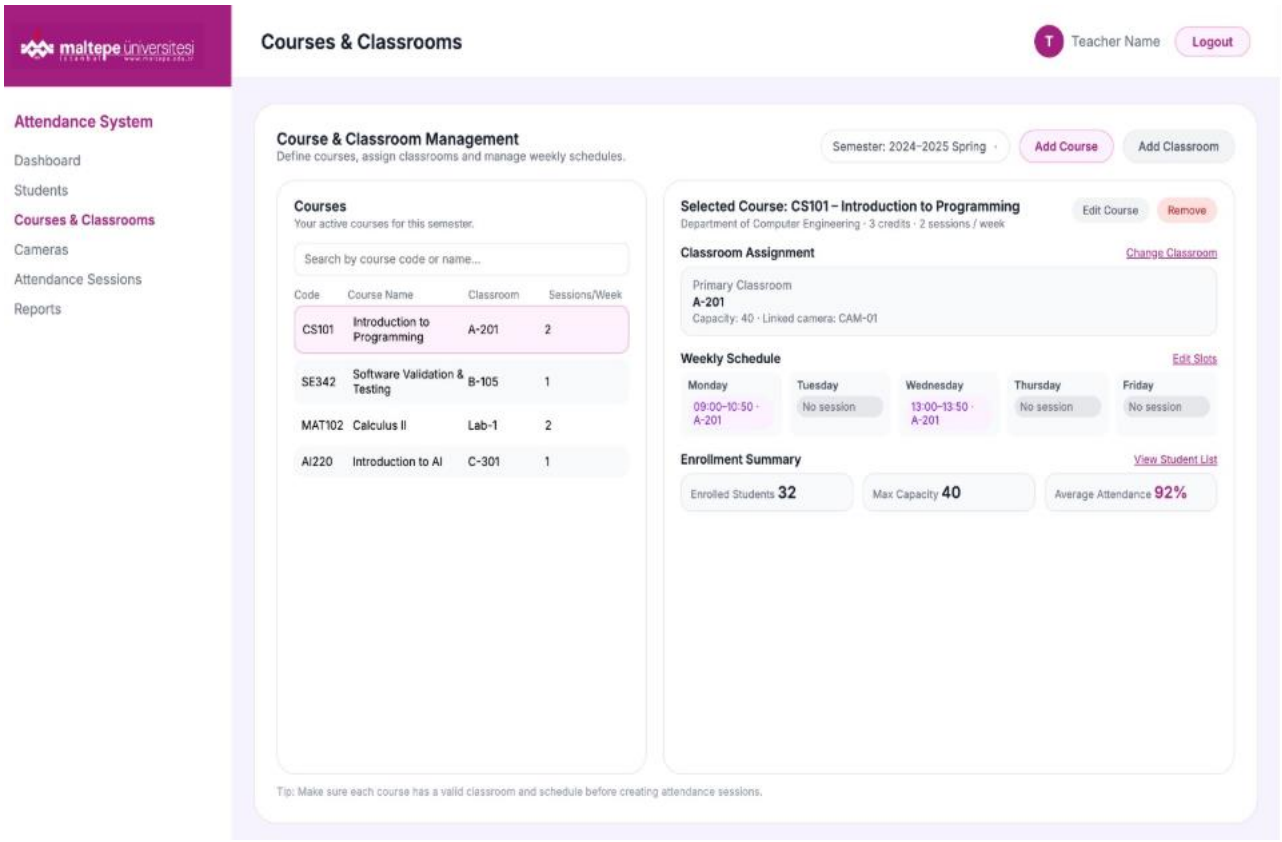

Description:
The Teacher Dashboard serves as the central control panel for instructors, providing quick access to all core features of the Smart Attendance System. From this interface, teachers can navigate to student management, course and classroom settings, camera setup, attendance sessions, and reporting tools. The dashboard offers an organized overview that helps instructors efficiently manage daily academic tasks and monitor classroom activities.

# 5.3 Students Page




Description:
The Students Page allows instructors to view and manage all registered students in the system. It provides a searchable list of students and displays detailed information—such as program, student ID, and attendance performance—for the selected student. This interface helps teachers quickly access student data and monitor academic participation.

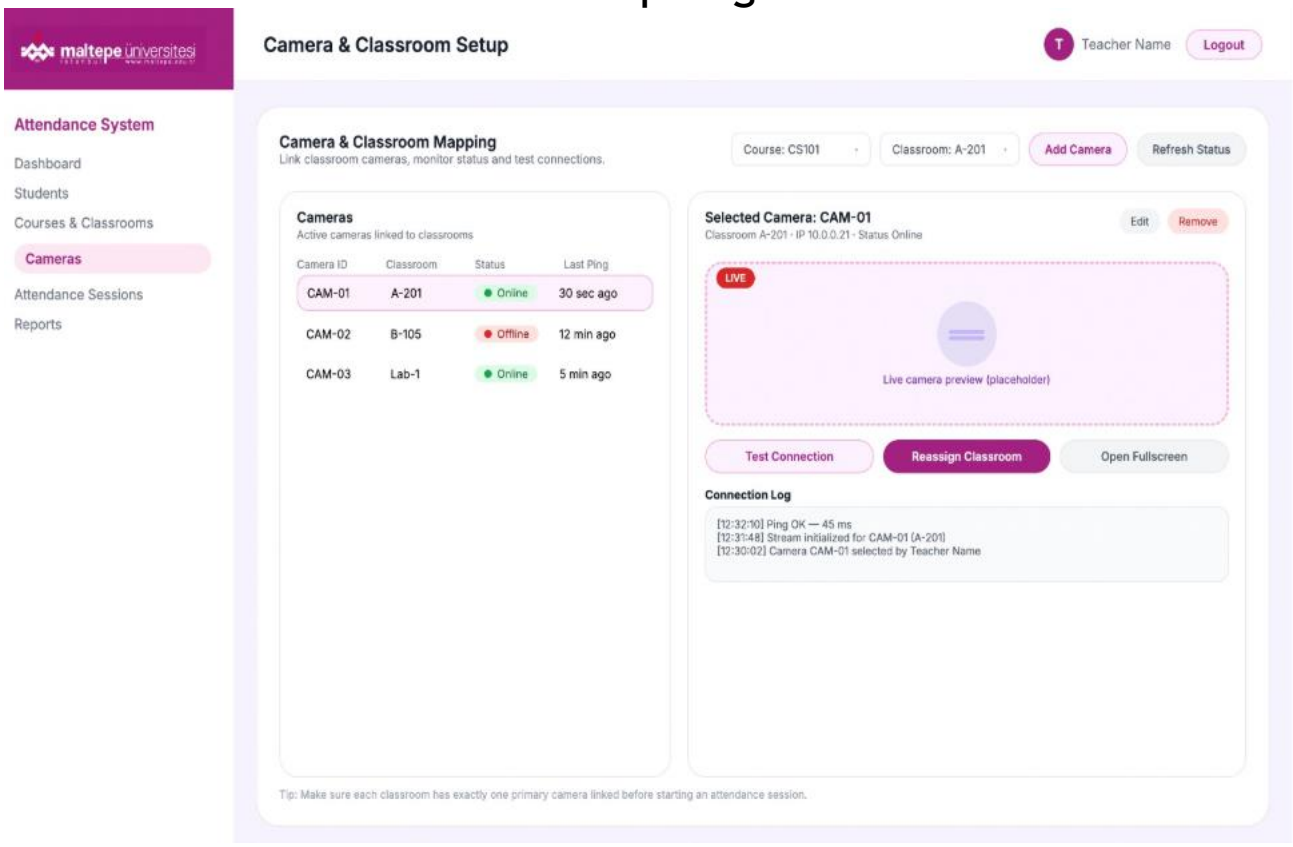
# 5.4 Courses & Classrooms Page




Description:

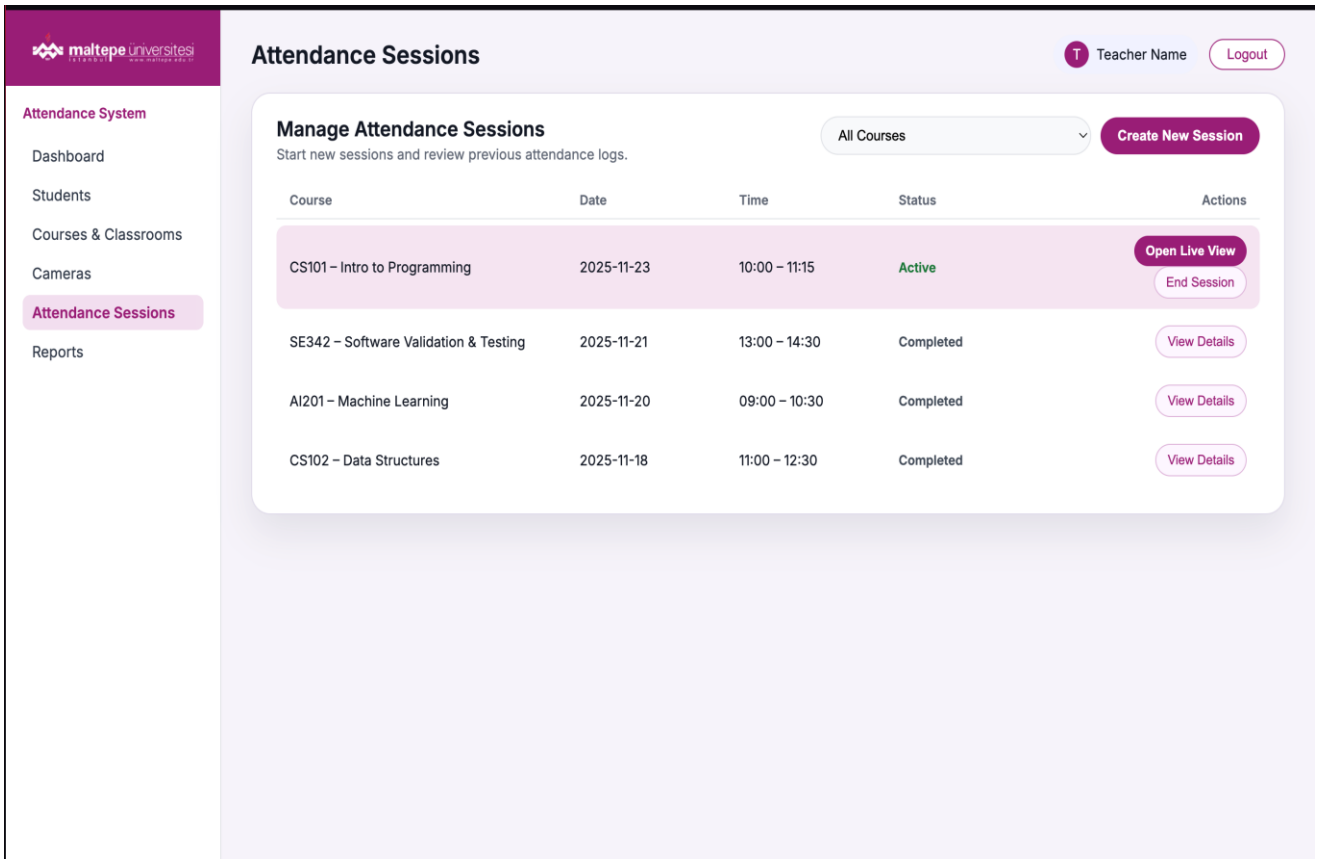
This page allows teachers to manage their courses, assign classrooms, and view weekly schedules. It provides a clear overview of course details, classroom assignments, and student enrollment, helping instructors organize their teaching activities efficiently.

# 5.5 Camera & Classroom Setup Page



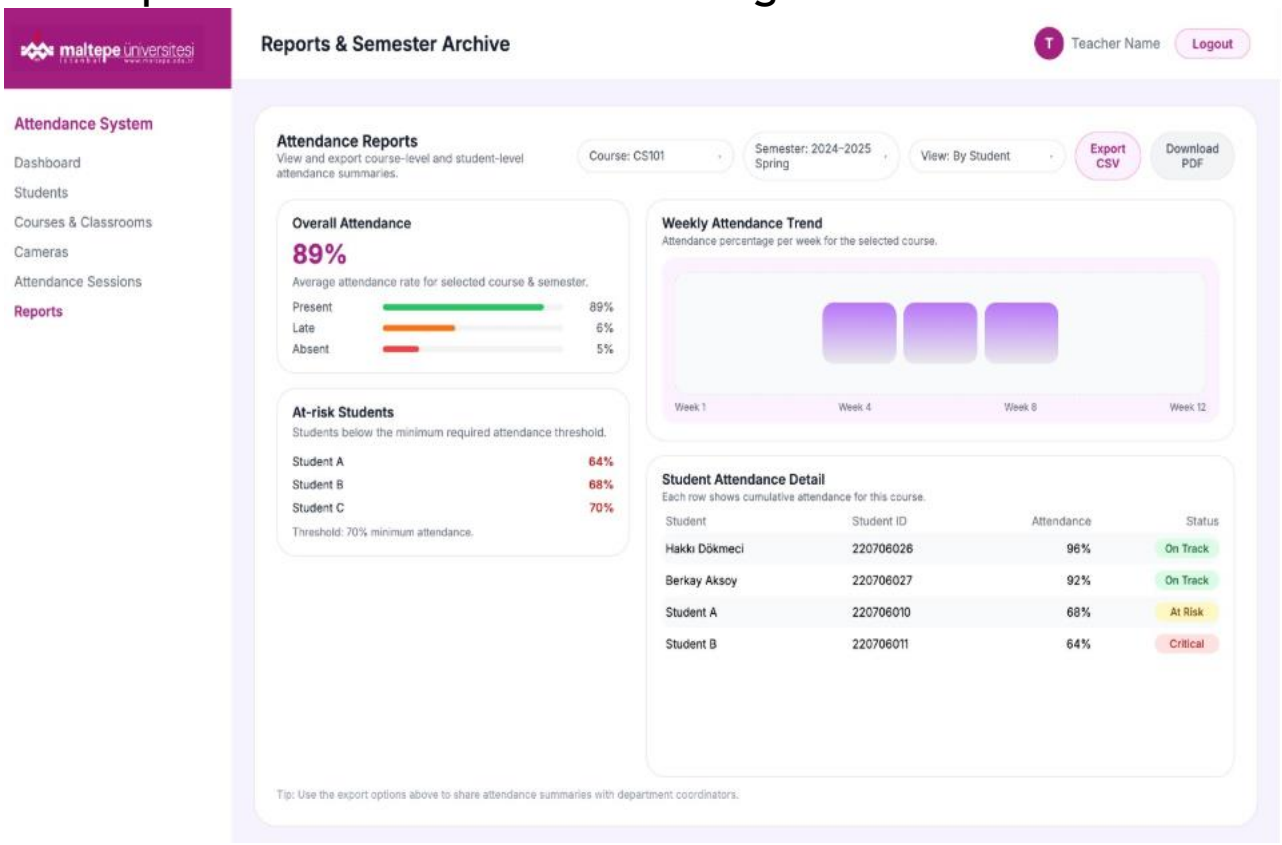

Description:
This page allows teachers to manage classroom cameras by monitoring their status, reviewing live previews, and assigning cameras to specific classrooms. It provides tools for testing connections, updating camera links, and viewing real-time activity logs to ensure reliable attendance recording.

# 5.6 Attendance Sessions Page




Description:
The Attendance Sessions Page allows instructors to start new attendance sessions, monitor active sessions, and review completed ones. It provides a structured list of all sessions with details such as course name, date, time, and session status. Teachers can quickly open live camera views, end active sessions, or review past attendance records through this interface.

# 5.7 Reports & Semester Archive Page



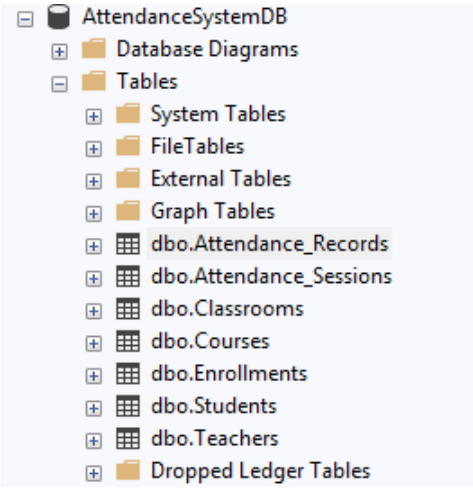

Description:
This page provides detailed attendance analytics for each course and semester. Teachers can view overall attendance rates, weekly attendance trends, at-risk students, and individual student records. It also offers export options such as CSV and PDF for sharing attendance summaries with academic departments.

# 6. Database development

The database layer of the system was implemented based on the previously designed ERD (Entity Relationship Diagram). All entities, attributes, relationships, and constraints defined in the ERD were successfully translated into a fully normalized relational database structure using Microsoft SQL Server. The final schema includes seven core tables—Students, Teachers, Courses, Classrooms, Enrollments, Attendance_Sessions, and Attendance_Records—ensuring data integrity, scalability, and support for the system's functional requirements.

## 6.1 Database Storage and Management Environment

The database structure designed based on the ERD was successfully created and stored in the Microsoft SQL Server (MSSQL) environment. As shown in the figure, all related tables are organized under the **AttendanceSystemDB** database, including Students, Teachers, Courses, Classrooms, Enrollments, Attendance_Sessions, and Attendance_Records. These tables are managed within SQL Server to ensure data consistency, relational integrity, secure storage, and efficient query execution throughout the system.

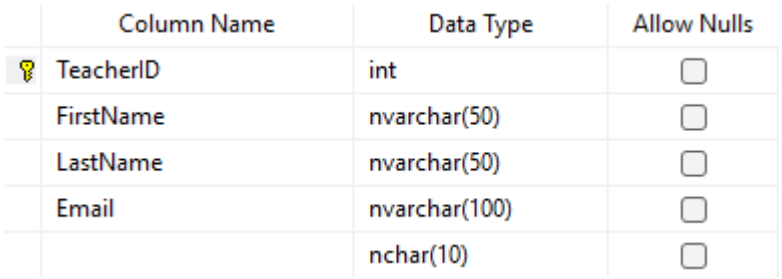



## 6.2 Database Tables Description

This section provides a detailed explanation of all database tables created as part of the attendance management system. Each table has been implemented based on the finalized Entity Relationship Diagram (ERD), ensuring proper normalization, data integrity, and relational consistency. The tables collectively support user management, course assignment, enrollment tracking, session scheduling, and attendance recording processes within the system.

## Teachers Table

Stores mandatory personal and contact information for instructors responsible for teaching courses.

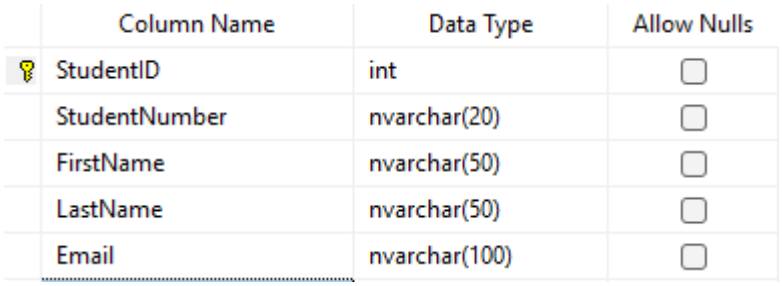


| Column Name | Data Type | Allow Nulls |
|---|---|---|
| TeacherID | int | ☐ |
| FirstName | nvarchar(50) | ☐ |
| LastName | nvarchar(50) | ☐ |
| Email | nvarchar(100) | ☐ |
| | nchar(10) | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `TeacherID` | `INT` | Primary Key, unique identifier |
| `FirstName` | `NVARCHAR(50)` | Instructor's first name |
| `LastName` | `NVARCHAR(50)` | Instructor's last name |
| `Email` | `NVARCHAR(100)` | Instructor's unique email address |

## Students Table

Stores academic identity and personal contact information for enrolled students.

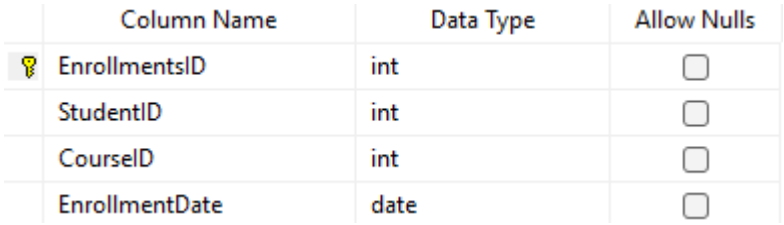


| Column Name | Data Type | Allow Nulls |
|---|---|---|
| StudentID | int | ☐ |
| StudentNumber | nvarchar(20) | ☐ |
| FirstName | nvarchar(50) | ☐ |
| LastName | nvarchar(50) | ☐ |
| Email | nvarchar(100) | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `StudentID` | `INT` | Primary Key |
| `StudentNumber` | `NVARCHAR(20)` | Unique institutional student number |
| `FirstName` | `NVARCHAR(50)` | Student's first name |
| `LastName` | `NVARCHAR(50)` | Student's last name |
| `Email` | `NVARCHAR(100)` | Student's email, must be unique |

## Enrollments Table

Represents the **bridge table** managing the Many-to-Many relationship between Students and Courses.

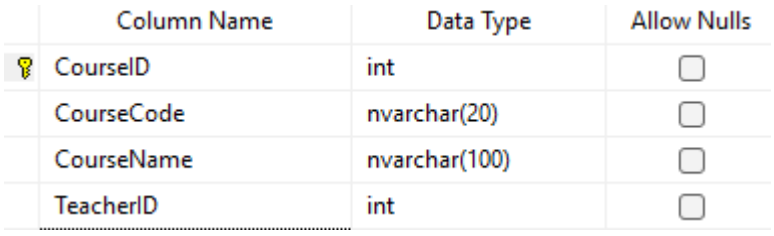


| Column Name | Data Type | Allow Nulls |
|---|---|---|
| EnrollmentsID | int | ☐ |
| StudentID | int | ☐ |
| CourseID | int | ☐ |
| EnrollmentDate | date | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `EnrollmentsID` | `INT` | Primary Key |
| `StudentID` | `INT` | Foreign Key referencing `Students` |
| `CourseID` | `INT` | Foreign Key referencing `Courses` |
| `EnrollmentDate` | `DATE` | Course enrollment date |

## Courses Table

Defines institution courses and links each one to the assigned teacher.

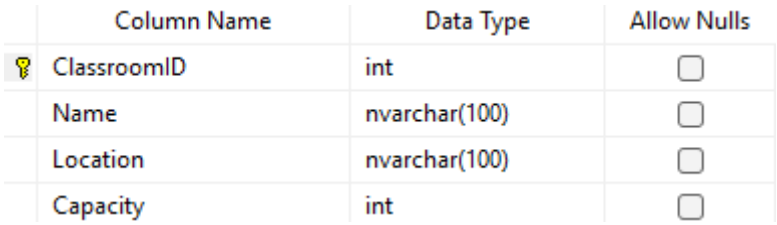


| Column Name | Data Type | Allow Nulls |
|---|---|---|
| CourseID | int | ☐ |
| CourseCode | nvarchar(20) | ☐ |
| CourseName | nvarchar(100) | ☐ |
| TeacherID | int | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `CourseID` | `INT` | Primary Key |
| `CourseCode` | `NVARCHAR(20)` | Unique course identifier |
| `CourseName` | `NVARCHAR(100)` | Full course title |
| `TeacherID` | `INT` | Foreign Key referencing `Teachers` |

## Classrooms Table

Stores physical classroom information used for scheduling teaching sessions.

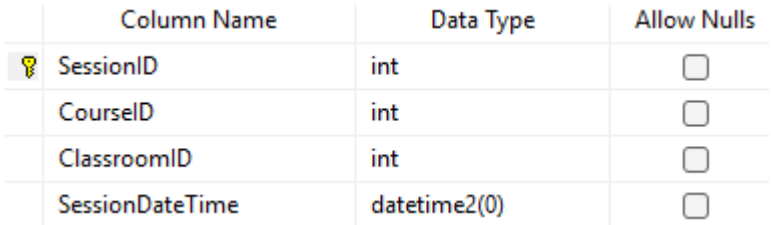


| Column Name | Data Type | Allow Nulls |
|---|---|---|
| ClassroomID | int | ☐ |
| Name | nvarchar(100) | ☐ |
| Location | nvarchar(100) | ☐ |
| Capacity | int | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `ClassroomID` | `INT` | Primary Key |
| `Name` | `NVARCHAR(100)` | Classroom title/label |
| `Location` | `NVARCHAR(100)` | Physical building/room identification |
| `Capacity` | `INT` | Maximum number of students allowed |

## 6-Attendance_Sessions Table

Records scheduled course sessions, connecting a lesson, location, and scheduled time.

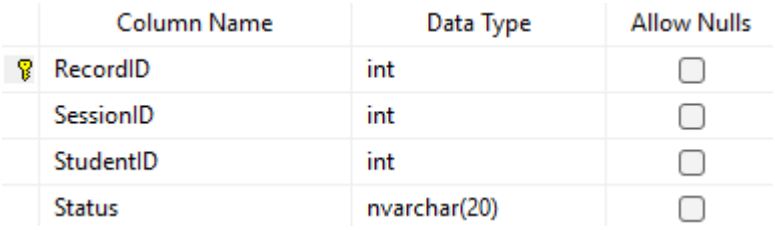


| Column Name | Data Type | Allow Nulls |
|---|---|---|
| SessionID | int | ☐ |
| CourseID | int | ☐ |
| ClassroomID | int | ☐ |
| SessionDateTime | datetime2(0) | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `SessionID` | `INT` | Primary Key |
| `CourseID` | `INT` | Foreign Key referencing `Courses` |
| `ClassroomID` | `INT` | Foreign Key referencing `Classrooms` |
| `SessionDateTime` | `DATETIME2(0)` | Exact start time of the session |

## 7- Attendance_Records Table

Stores attendance results for each student and each session.



| Column Name | Data Type | Allow Nulls |
|---|---|---|
| RecordID | int | ☐ |
| SessionID | int | ☐ |
| StudentID | int | ☐ |
| Status | nvarchar(20) | ☐ |

| Column | Data Type | Purpose |
|---|---|---|
| `RecordID` | `INT` | Primary Key |
| `SessionID` | `INT` | Foreign Key referencing `Attendance_Sessions` |
| `StudentID` | `INT` | Foreign Key referencing `Students` |
| `Status` | `NVARCHAR(20)` | Attendance status (`Present`, `Absent`, `Late`) |