



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

1. Main Sections (EPICs)

- 1.1 EPIC 1 - System Requirements & Analysis
- 1.2 EPIC 2 - Database Analysis & Setup
- 1.3 EPIC 3 - Student Registration & Photo Process
- 1.4 EPIC 4 - Course & Classroom Management
- 1.5 EPIC 5 - Camera & Classroom Setup
- 1.6 EPIC 6 - Attendance Session & Recognition Flow
- 1.7 EPIC 7 - Teacher Approval & Manual Adjustment
- 1.8 EPIC 8 - Reporting & Semester Archive
- 1.9 EPIC 9 - Privacy, Access, and Data Retention Policies
- 1.10 EPIC 10 - Testing & Evaluation
- 1.11 EPIC 11 - Deployment & Maintenance

2. Database Design

- 2.1 ERD Description

3. Sequence Diagrams

- 3.1 Student Registration Sequence
- 3.2 Camera Setup Sequence
- 3.3 Attendance Session Sequence

4. Use Case Diagram

- 4.1 Use-Case Diagram Description

5. Page Designs

- 5.1 Login Page
- 5.2 Teacher Dashboard Page
- 5.3 Students Page
- 5.4 Courses & Classrooms Page
- 5.5 Camera & Classroom Setup Page
- 5.6 Attendance Sessions Page
- 5.7 Reports & Semester Archive Page

6. Database Development

- 6.1 Database Storage and Management Environment
- 6.2 Database Tables Description

Classroom Automatic Attendance System

Requirements Analysis Document - Main Sections (EPICs)

EPIC 1 - System Requirements & Analysis

EPIC 2 - Database Analysis & Setup

EPIC 3 - Student Registration & Photo Process

EPIC 4 - Course & Classroom Management

EPIC 5 - Camera & Classroom Setup

EPIC 6 - Attendance Session & Recognition Flow

EPIC 7 - Teacher Approval & Manual Adjustment

EPIC 8 - Reporting & Semester Archive

EPIC 9 - Privacy, Access, and Data Retention Policies

EPIC 10 - Testing & Evaluation

EPIC 11- Deployment & Maintenance

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

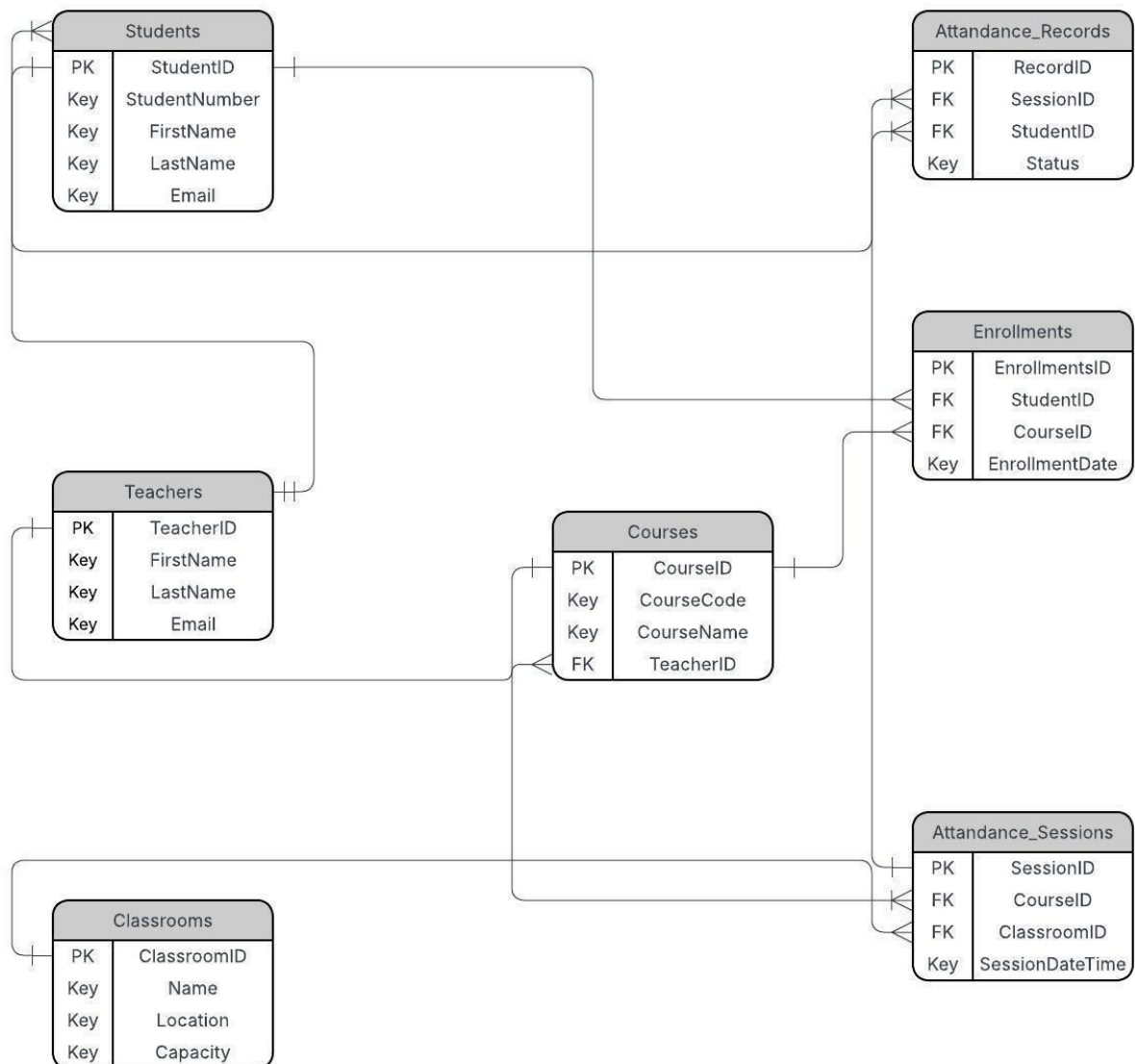
Story: System Deployment & Environment Setup

-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

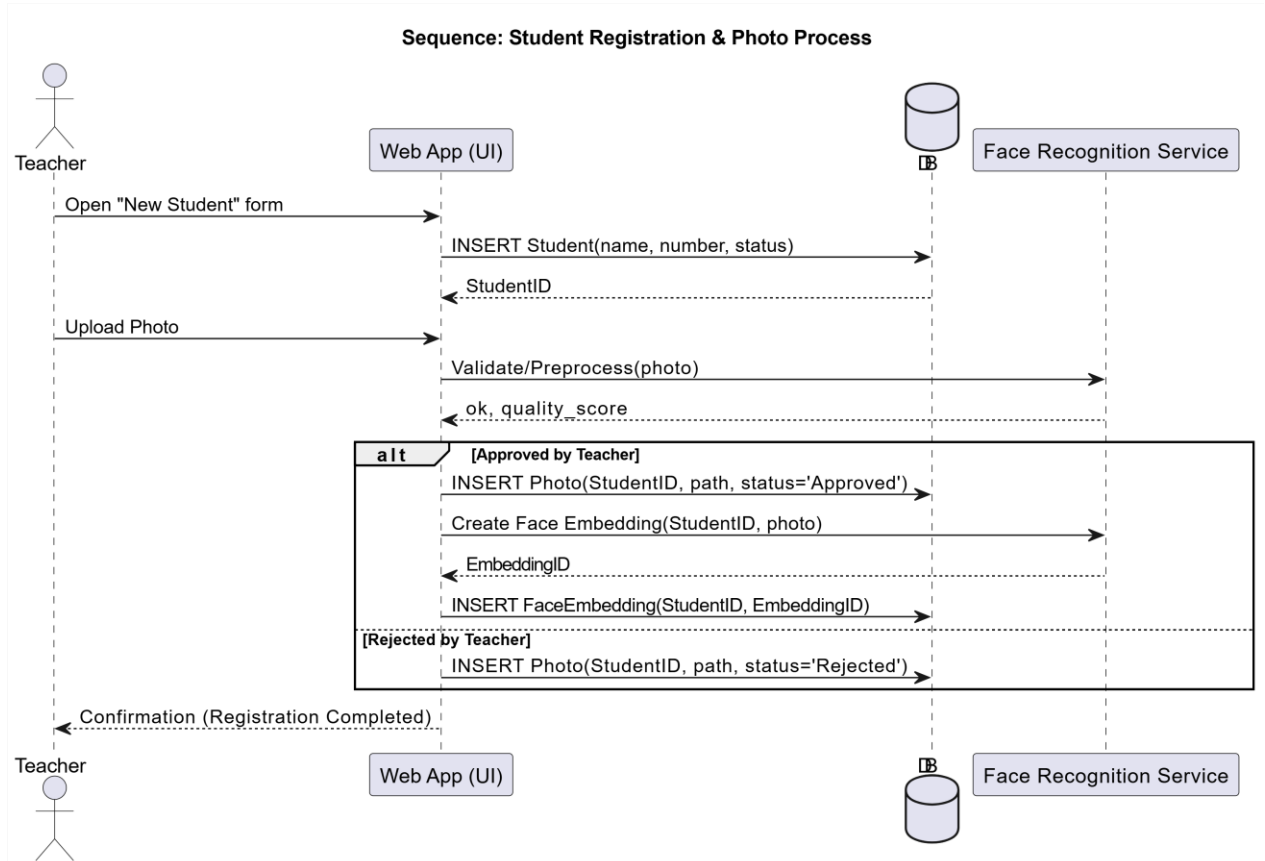
- Represents scheduled attendance sessions for a course.
- Linked to:
 - One **Course**
 - One **Classroom**
- Each session produces multiple **Attendance_Records**.

7. Attendance_Records

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

3. Sequence Diagram

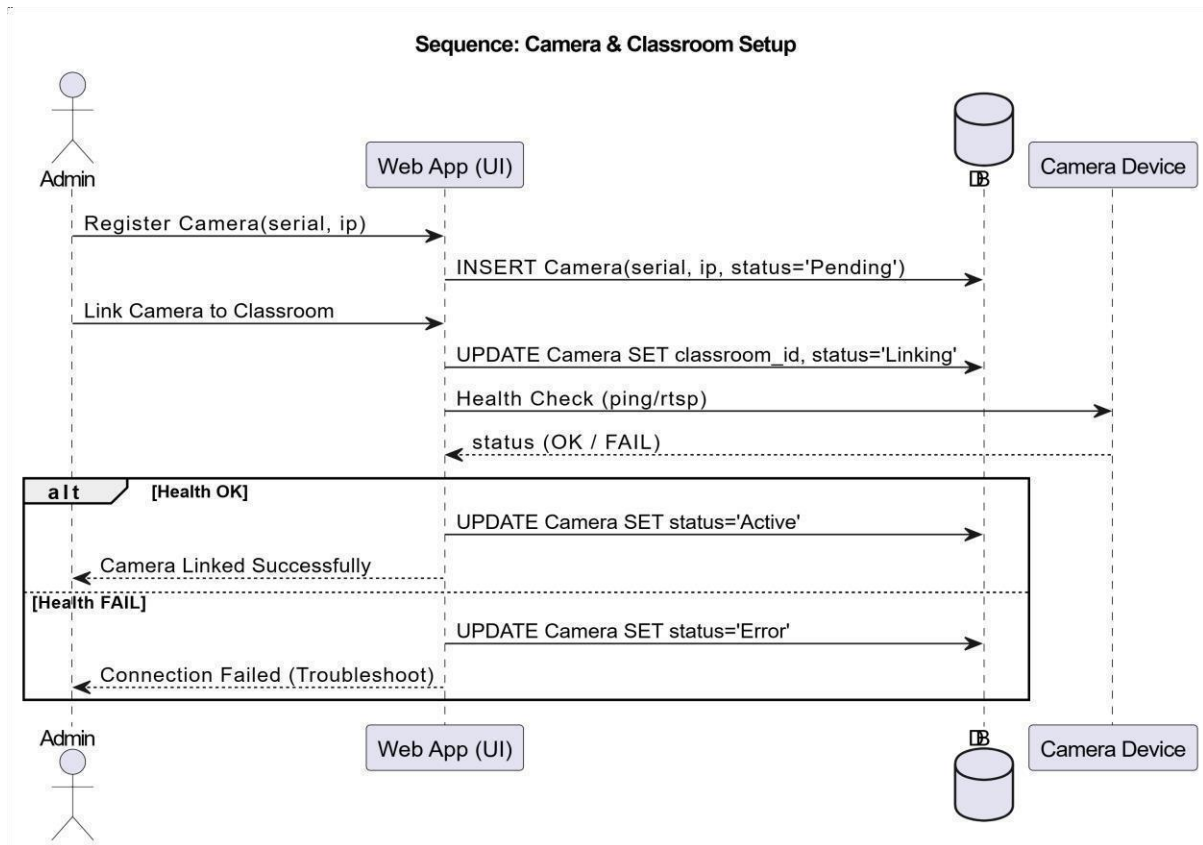
3.1 Sequence Diagram (Student Registration & Photo Process)



Description:

- This workflow 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.

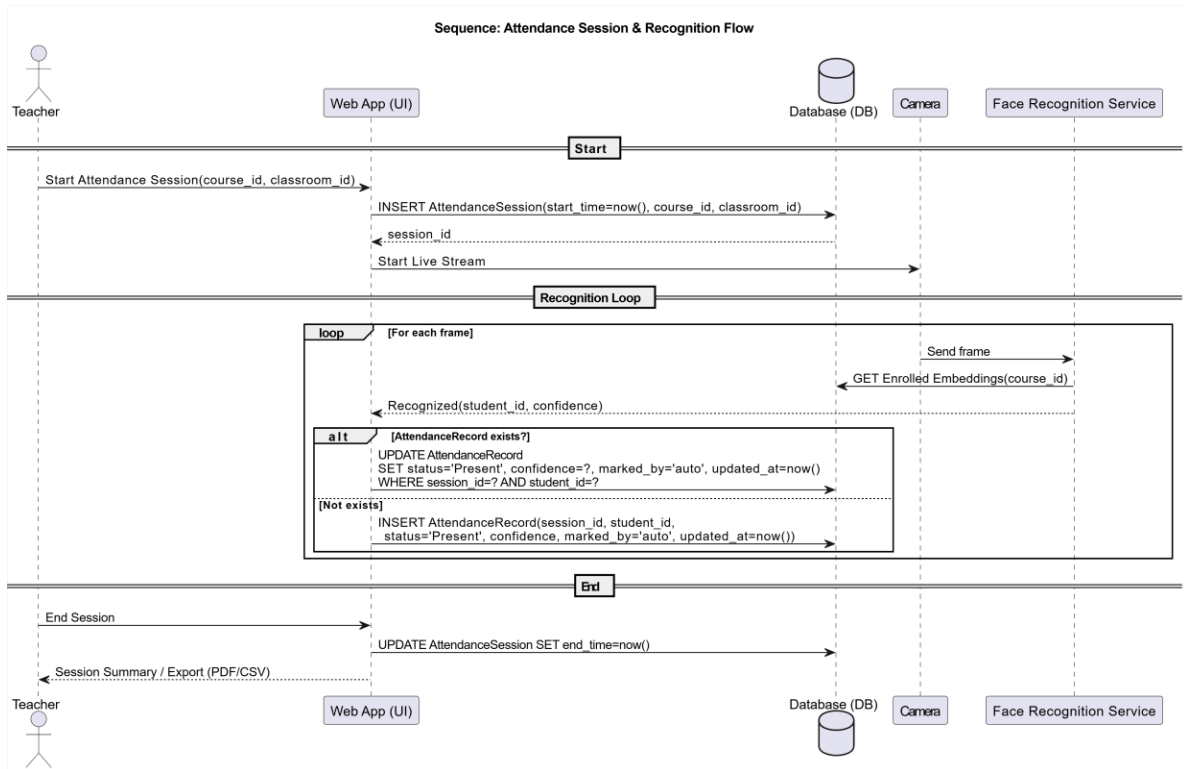
3.2 Sequence Diagram (Camera & Classroom Setup)



Description:

- This workflow 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 (Attendance Session & Recognition Flow)

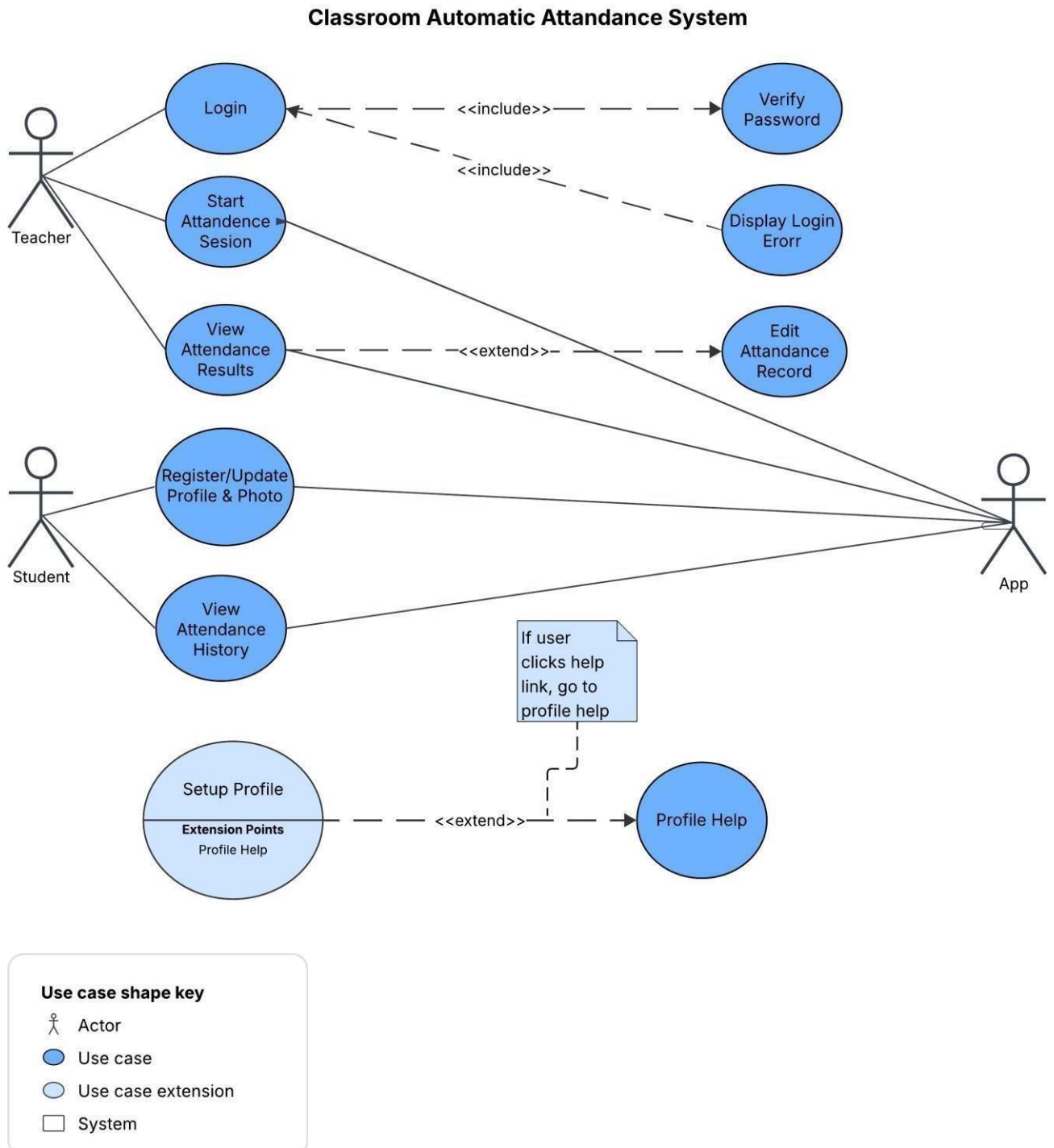


Description:

- This is the main **operational phase** – where the teacher starts a session, the camera streams video, and the recognition service marks attendance.
- It depends on 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.

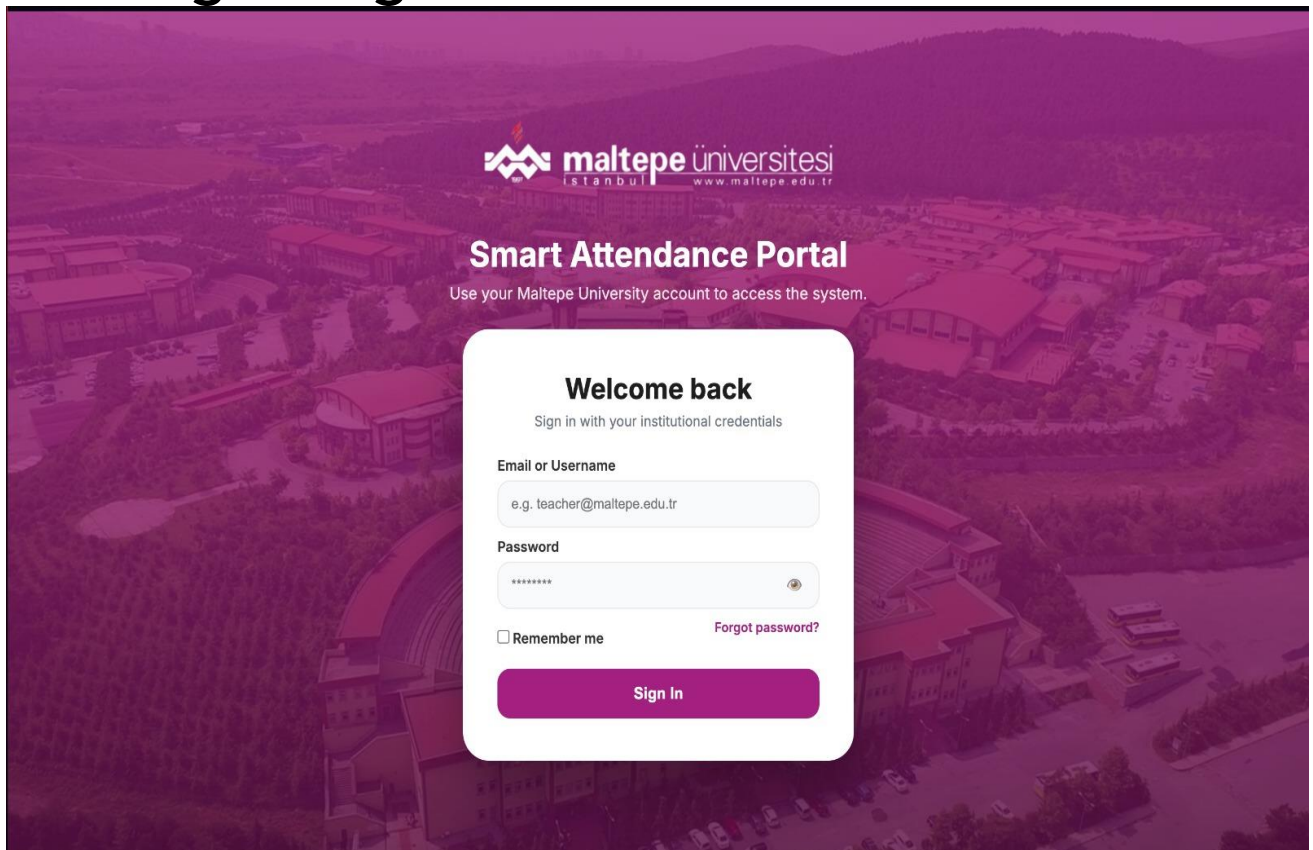
<<extend>> — Role in the Diagram

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



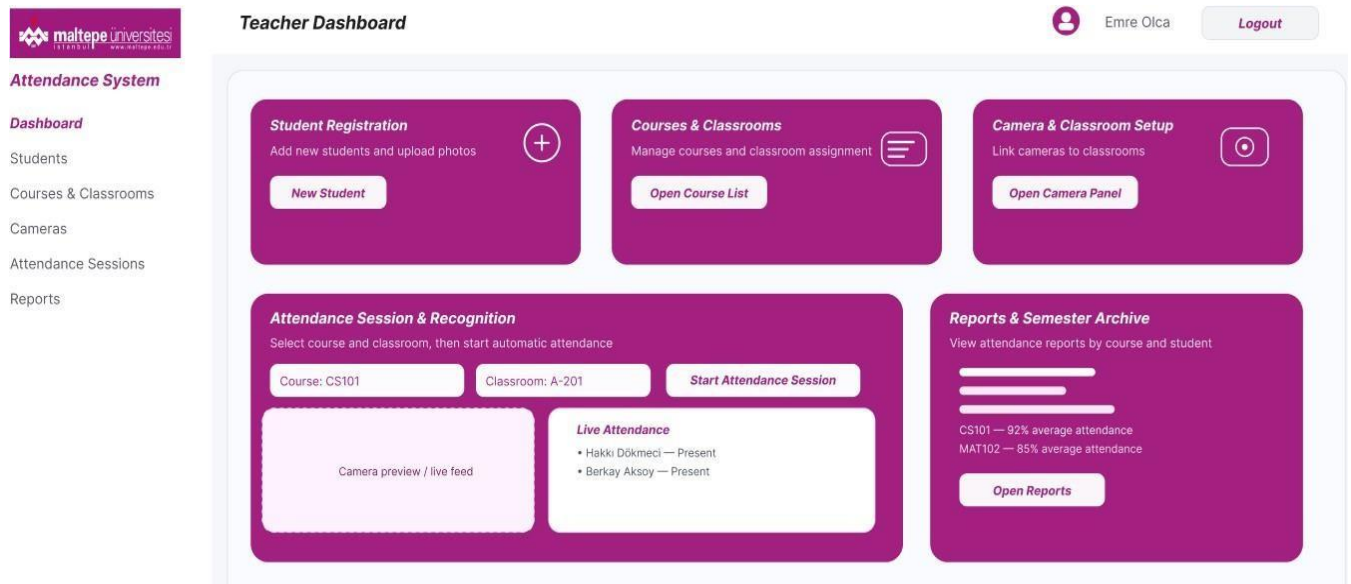
The screenshot shows the login page of the Maltepe University Smart Attendance Portal. The background is a purple-tinted aerial view of the university campus. At the top center is the Maltepe University logo, which includes a stylized building icon and the text "maltepe üniversitesi istanbul www.maltepe.edu.tr". Below the logo, the title "Smart Attendance Portal" is displayed in white, followed by the instruction "Use your Maltepe University account to access the system." in a smaller font. The main login form is a white rounded rectangle with a purple border. It features a "Welcome back" heading and a sub-instruction "Sign in with your institutional credentials". The form contains two input fields: "Email or Username" with a placeholder "e.g. teacher@maltepe.edu.tr" and "Password" with a masked input "*****" and a toggle icon. Below these fields are a "Remember me" checkbox and a "Forgot password?" link. A prominent purple "Sign in" button is at the bottom of the form.

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.

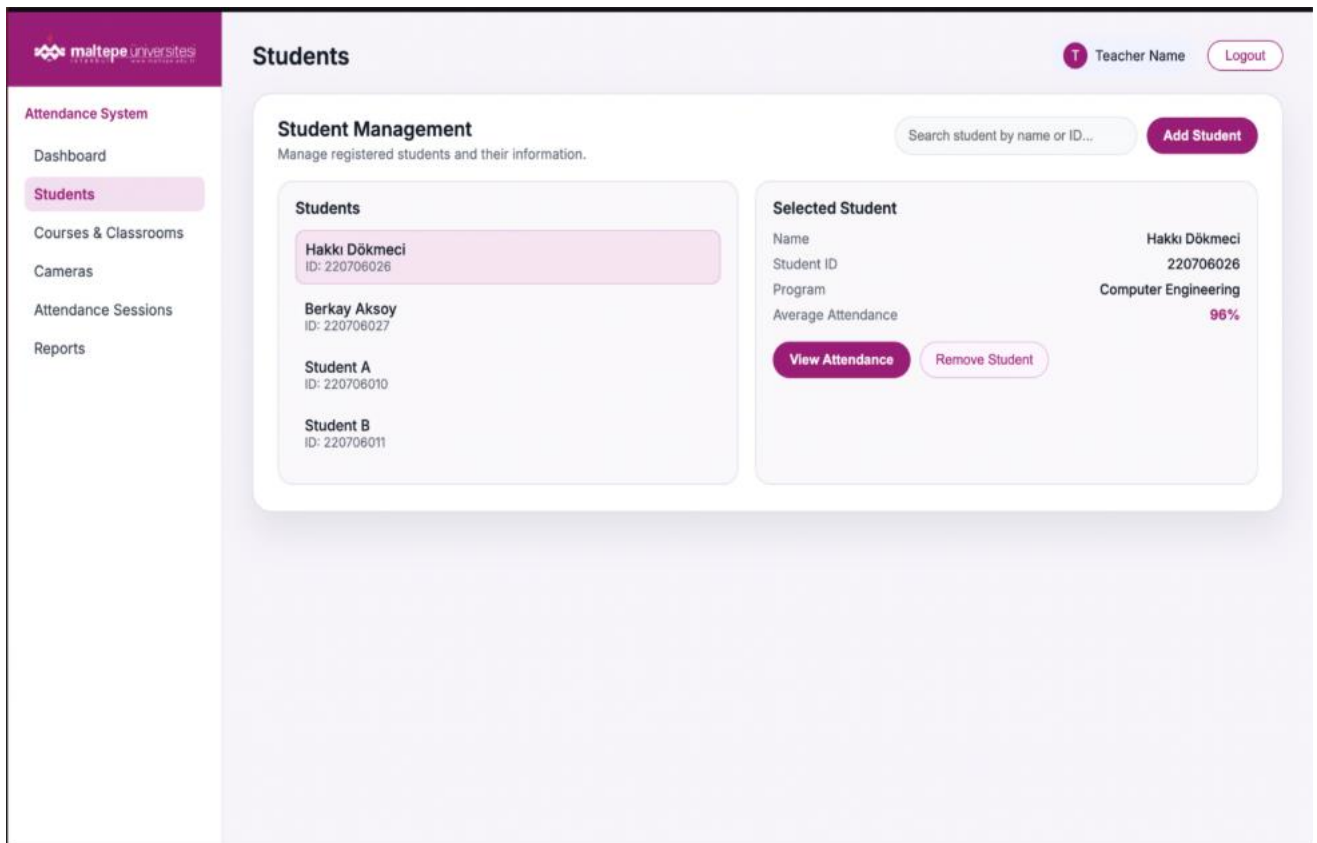
5.2 Teacher Dashboard Page



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

maltepe universitesi
1982-2025

Courses & Classrooms

Teacher Name Logout

Semester: 2024-2025 Spring Add Course Add Classroom

Course & Classroom Management

Define courses, assign classrooms and manage weekly schedules.

Courses

Your active courses for this semester.

Search by course code or name...

Code	Course Name	Classroom	Sessions/Week
CS101	Introduction to Programming	A-201	2
SE342	Software Validation & Testing	B-105	1
MAT102	Calculus II	Lab-1	2
AI220	Introduction to AI	C-301	1

Selected Course: CS101 – Introduction to Programming

Department of Computer Engineering · 3 credits · 2 sessions / week

Edit Course Remove

Classroom Assignment

Change Classroom

Primary Classroom
A-201
Capacity: 40 · Linked camera: CAM-01

Weekly Schedule

Edit Slots

Monday	Tuesday	Wednesday	Thursday	Friday
09:00-10:50 · A-201	No session	13:00-13:50 · A-201	No session	No session

Enrollment Summary

View Student List

Enrolled Students **32** Max Capacity **40** Average Attendance **92%**

Tip: Make sure each course has a valid classroom and schedule before creating attendance sessions.

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

The screenshot shows the 'Camera & Classroom Setup' page. On the left is a sidebar with the 'Attendance System' logo and navigation links: Dashboard, Students, Courses & Classrooms, Cameras (highlighted), Attendance Sessions, and Reports. The main header includes the page title 'Camera & Classroom Setup', a user profile 'Teacher Name', and a 'Logout' button. The main content area is divided into two panels. The left panel, 'Camera & Classroom Mapping', lists active cameras with columns for Camera ID, Classroom, Status, and Last Ping. The right panel, 'Selected Camera: CAM-01', shows a live camera preview placeholder and a connection log. At the bottom, a tip states: 'Tip: Make sure each classroom has exactly one primary camera linked before starting an attendance session.'

Attendance System

Dashboard
Students
Courses & Classrooms
Cameras
Attendance Sessions
Reports

Camera & Classroom Setup Teacher Name Logout

Camera & Classroom Mapping
Link classroom cameras, monitor status and test connections.

Course: CS101 Classroom: A-201 Add Camera Refresh Status

Camera ID	Classroom	Status	Last Ping
CAM-01	A-201	Online	30 sec ago
CAM-02	B-105	Offline	12 min ago
CAM-03	Lab-1	Online	5 min ago

Selected Camera: CAM-01
Classroom A-201 · IP 10.0.0.21 · Status Online Edit Remove

LIVE

Live camera preview (placeholder)

Test Connection Reassign Classroom Open Fullscreen

Connection Log

- [12:32:10] Ping OK — 45 ms
- [12:31:48] Stream initialized for CAM-01 (A-201)
- [12:30:02] Camera CAM-01 selected by Teacher Name

Tip: Make sure each classroom has exactly one primary camera linked before starting an attendance session.

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

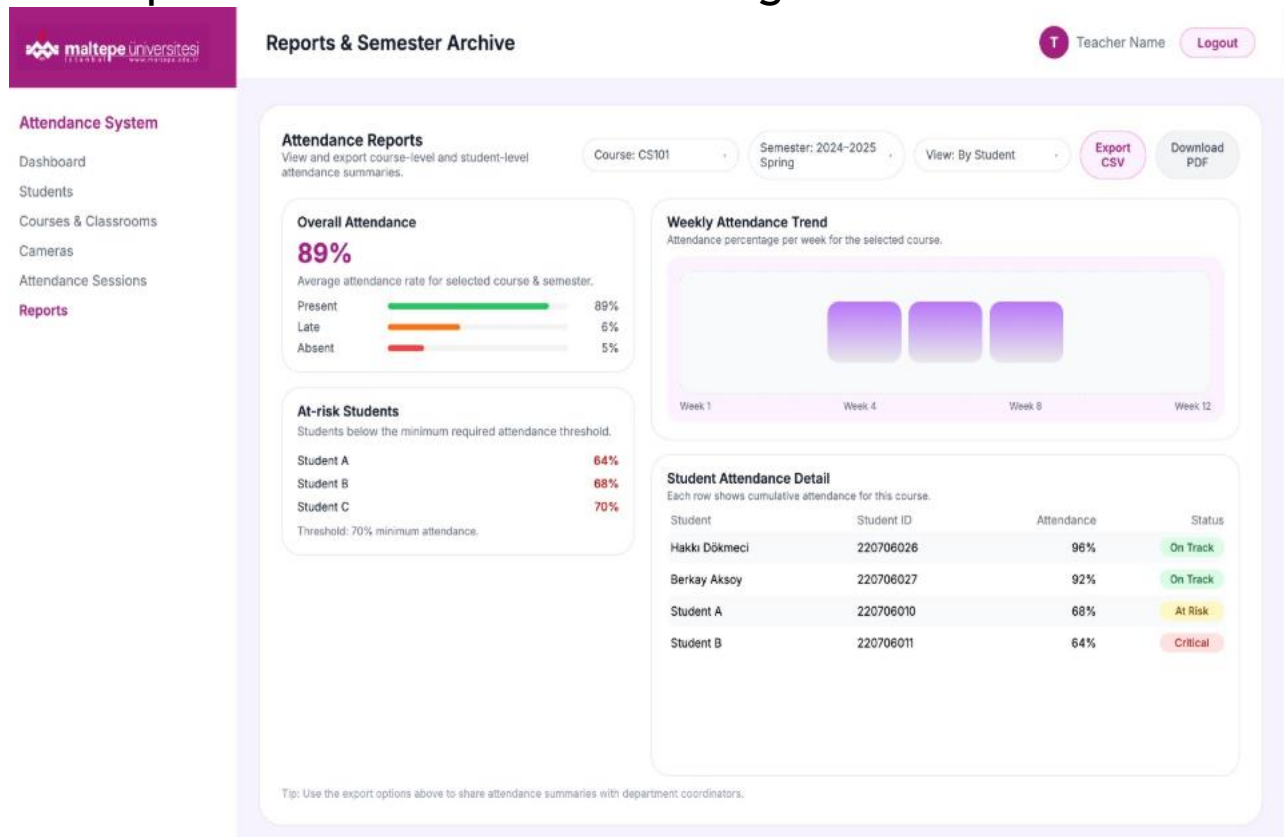
The screenshot shows the 'Attendance Sessions' page of the Maltepe University system. The page has a purple sidebar on the left with navigation links: 'Attendance System', 'Dashboard', 'Students', 'Courses & Classrooms', 'Cameras', 'Attendance Sessions' (highlighted), and 'Reports'. The main content area is titled 'Attendance Sessions' and includes a user profile 'Teacher Name' with a 'Logout' button. Below the title is a 'Manage Attendance Sessions' section with a subtitle 'Start new sessions and review previous attendance logs.' and a dropdown menu set to 'All Courses'. A 'Create New Session' button is located to the right. The main part of the page is a table with the following data:

Course	Date	Time	Status	Actions
CS101 - Intro to Programming	2025-11-23	10:00 - 11:15	Active	Open Live View End Session
SE342 - Software Validation & Testing	2025-11-21	13:00 - 14:30	Completed	View Details
AI201 - Machine Learning	2025-11-20	09:00 - 10:30	Completed	View Details
CS102 - Data Structures	2025-11-18	11:00 - 12:30	Completed	View Details

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:

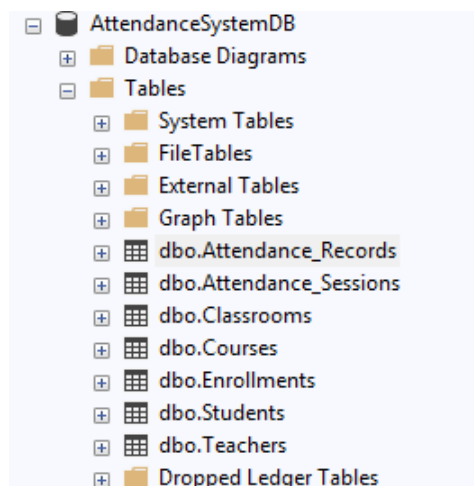
This page provides detailed attendance analytics for each course and semester. Teachers can view overall attendance rates, weekly attendance trends, at-risk students, and individual student records. It also offers export options such as CSV and PDF for sharing attendance summaries with academic departments.

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	<input type="checkbox"/>
	FirstName	nvarchar(50)	<input type="checkbox"/>
	LastName	nvarchar(50)	<input type="checkbox"/>
	Email	nvarchar(100)	<input type="checkbox"/>
		nchar(10)	<input type="checkbox"/>

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	<input type="checkbox"/>
	StudentNumber	nvarchar(20)	<input type="checkbox"/>
	FirstName	nvarchar(50)	<input type="checkbox"/>
	LastName	nvarchar(50)	<input type="checkbox"/>
	Email	nvarchar(100)	<input type="checkbox"/>

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	<input type="checkbox"/>
	StudentID	int	<input type="checkbox"/>
	CourseID	int	<input type="checkbox"/>
	EnrollmentDate	date	<input type="checkbox"/>

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	<input type="checkbox"/>
	CourseCode	nvarchar(20)	<input type="checkbox"/>
	CourseName	nvarchar(100)	<input type="checkbox"/>
	TeacherID	int	<input type="checkbox"/>

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	<input type="checkbox"/>
	Name	nvarchar(100)	<input type="checkbox"/>
	Location	nvarchar(100)	<input type="checkbox"/>
	Capacity	int	<input type="checkbox"/>

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	<input type="checkbox"/>
	CourseID	int	<input type="checkbox"/>
	ClassroomID	int	<input type="checkbox"/>
	SessionDateTime	datetime2(0)	<input type="checkbox"/>

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	<input type="checkbox"/>
	SessionID	int	<input type="checkbox"/>
	StudentID	int	<input type="checkbox"/>
	Status	nvarchar(20)	<input type="checkbox"/>

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)

