



MALTEPE UNIVERSITY

Faculty of Engineering and Natural Sciences - Software Engineering

SE 342

Software Validation and Testing

Instructor: Dr. Emre Atlıer Olca

Prepared by:

22 07 06 059 - Ceren Yaşar

22 07 06 028 - Mehmet Şenadlı

22 07 06 018 - Bilal Çifteci

Table of Contents

<i>Smart Attendance System</i>	3
1. Introduction.....	3
1.1 Project Purpose.....	3
1.2 Scope	3
2. Requirements.....	3
2.1. Functional Requirements	3
2.2. Non-Functional Requirements (NFR)	4
3. System Design	6
3.1 ERD (Entity Relationship Diagram)	6
3.2 Database Schema Design (PostgreSQL).....	7
3.3 UML Diagrams.....	9
3.3.1 Use Case Diagram.....	9
3.3.2 Sequence Diagrams.....	10
4. User Interface (UI) Design	13
▪ Log in Page	13
▪ Student Registration Screen	14
▪ Automatic Attendance Session	15
▪ Attendance Reports for Teacher.....	16
▪ Student Reports Panel	17
5. Project Plan / Methodology.....	18
5.1 Sprint 0: Software Requirement and Analysis	18
5.2 Sprint 1: Database Design and Management	18
5.3 Sprint 2: Student Management and Registration Module	18
5.4 Sprint 3: Camera Integration System	19
5.5 Sprint 4: Reporting and Admin Panel.....	19

Smart Attendance System

For detailed tasks and progress, see the Jira board: [!\[\]\(1d3a1175dd4902218e694b9c098adb83_img.jpg\) Jira Link](#)

For source code and repository structure, see the GitHub project: [!\[\]\(c507f772dba2b921f86777f01218e570_img.jpg\) GitHub Link](#)

1. Introduction

1.1 Project Purpose

This system aims to automate the attendance process in an educational institution. The process begins with the teacher adding student records along with their photos to the system. The integrated camera system then detects student faces in the classroom and compares them with the registered photos. When a successful match occurs, attendance is automatically recorded, and a “Attendance recorded” notification is generated in the system.

1.2 Scope

The automated attendance system is developed to accelerate the attendance process and minimize errors in educational institutions. The system includes the following features:

- Adding student records to the system along with their photos,
- Recognizing and matching student faces in the classroom using the camera,
- Automatically recording attendance upon successful matches,
- Providing real-time notifications to teachers and administrators about the attendance status.

2. Requirements

2.1. Functional Requirements

FR-1: Student Management and Registration Module (SCRUM-25)

- **FR-1.1:** The system must allow the creation of new student registrations. (SCRUM-1)
- **FR-1.2:** A student registration form (name, number, department, class, photo) must be created. (SCRUM-5, 6)
- **FR-1.3:** Validation rules must be added for form fields. (SCRUM-7)
- **FR-1.4:** A student photo upload function must be developed. (SCRUM-8)

- **FR-1.5:** Uploaded photos must be stored in the system (database or file system). (SCRUM-10)
- **FR-1.6:** Backend integration for photo/file upload must be implemented. (SCRUM-9)
- **FR-1.7:** A message must be shown to the user after successful registration. (SCRUM-11)

FR-2: Automatic Face Recognition and Attendance

- **FR-2.1:** The system must have a camera integration system. (SCRUM-26, 4)
- **FR-2.2:** The system must be able to perform automatic face recognition and attendance tracking via camera. (SCRUM-2)
- **FR-2.3:** A face recognition and matching system must be established. (SCRUM-12)
- **FR-2.4:** The system must have an algorithm that compares the face from the camera with existing photos registered in the system. (SCRUM-13)
- **FR-2.5:** An accuracy threshold for matching must be determined. (SCRUM-14)
- **FR-2.6:** After a successful match, the attendance record must be created automatically. (SCRUM-15)
- **FR-2.7:** A real-time "Attendance taken" message must be displayed when attendance is recorded. (SCRUM-16)

FR-3: Reporting and Admin Panel (SCRUM-27)

- **FR-3.1:** There must be a reporting and management interface for attendance data. (SCRUM-3)
- **FR-3.2:** A page displaying daily attendance data must be created. (SCRUM-18)
- **FR-3.3:** The reporting page must have filtering and sorting features. (SCRUM-19)
- **FR-3.4:** Reports must be filterable by name. (SCRUM-20)
- **FR-3.5:** Reports must be sortable by department/class. (SCRUM-21)
- **FR-3.6:** The system must provide a warning for unrecognized students and allow manual attendance entry. (SCRUM-17)
- **FR-3.7:** The system must include a module for editing incorrect attendance records. (SCRUM-22)
- **FR-3.8:** Frontend integration for real-time updates in the admin panel must be implemented. (SCRUM-23)

2.2. Non-Functional Requirements (NFR)

NFR-1: Database and Infrastructure

- **NFR-1.1:** A database engine must be selected and installed. (SCRUM-34)
- **NFR-1.2:** Database setup and configuration must be performed. (SCRUM-33)
- **NFR-1.3:** A database connection configuration (connection string) must be defined. (SCRUM-35)
- **NFR-1.4:** An ORM (SQLAlchemy, Prisma, Django ORM, etc.) must be selected and developed. (SCRUM-38, 37)
- **NFR-1.5:** Database model classes (entities) must be created. (SCRUM-39)

NFR-2: Security

- **NFR-2.1:** Database security and access management must be ensured. (SCRUM-40)
- **NFR-2.2:** User roles must be defined and authorization implemented. (SCRUM-41)
- **NFR-2.3:** Encryption and access policies for sensitive data must be applied. (SCRUM-42)
- **NFR-2.4:** The system must have connection security (e.g., SSL). (SCRUM-43)
- **NFR-2.5:** Database access tests must be conducted. (SCRUM-36)

NFR-3: Performance and Maintenance

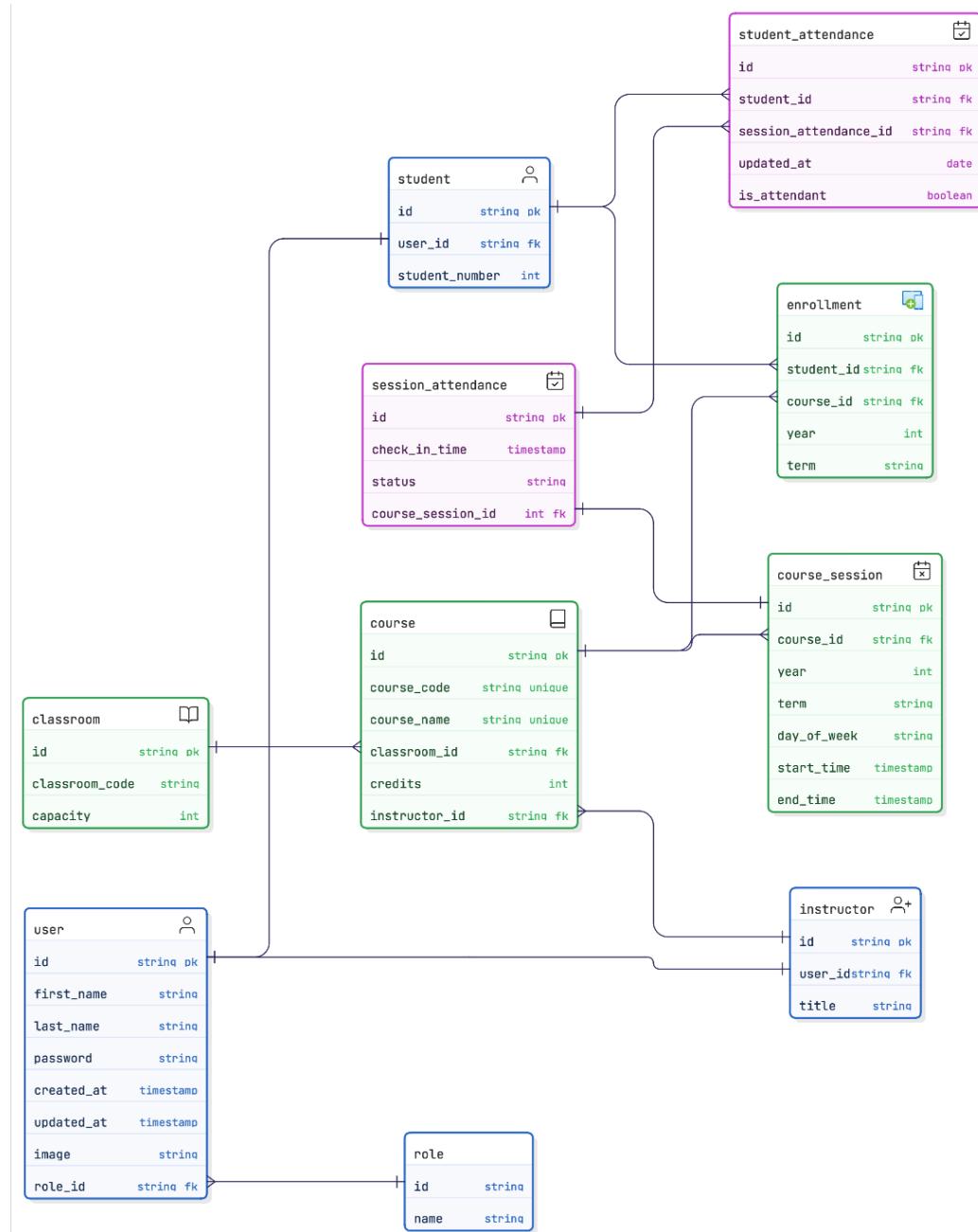
- **NFR-3.1:** Database backup and performance management must be planned. (SCRUM-44)
- **NFR-3.2:** An automatic backup plan must be created. (SCRUM-45)
- **NFR-3.3:** Query performance analyses must be conducted. (SCRUM-47)
- **NFR-3.4:** Necessary indexing strategies must be determined. (SCRUM-46)

NFR-4: Analysis and Documentation

- **NFR-4.1:** Requirements analysis and documentation for the project must be completed. (SCRUM-48, 49, 50, 51, 52, 53, 54)
- **NFR-4.2:** A requirements document (SRS) must be created. (SCRUM-55)

3. System Design

3.1 ERD (Entity Relationship Diagram)



3.2 Database Schema Design (PostgreSQL)

The database was created in PostgreSQL based on a Crow's Foot ERD to support an automated attendance system.

The User table is the core entity, and each user is linked to exactly one Student or Instructor, forming a strict one-to-one relationship through unique foreign keys.

The academic structure consists of Course, Classroom, and Instructor tables.

Each course is assigned to one instructor and one classroom, and course codes and names are kept unique.

Weekly lectures are represented through Course Session, which stores day and time information.

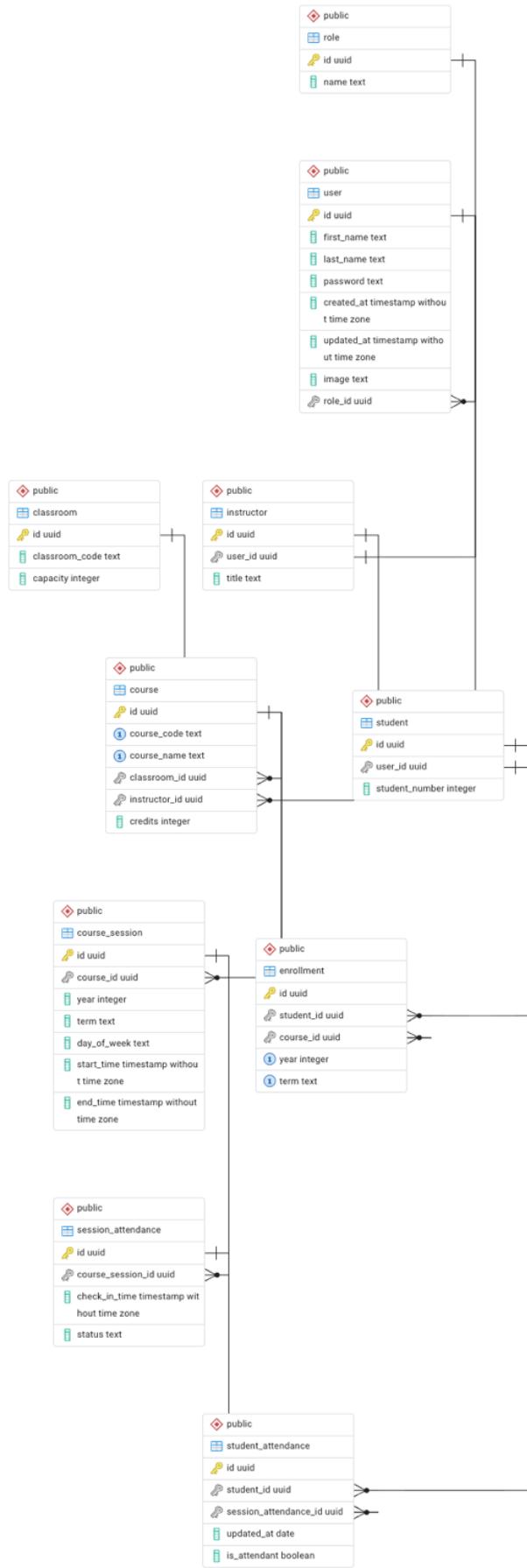
Attendance is handled with Session Attendance for each session and Student Attendance for each student's attendance status.

A unique constraint prevents duplicate attendance entries for the same student in the same session.

Students' course registrations are stored in Enrollment, which ensures that a student cannot register for the same course twice in the same term.

All primary keys, foreign keys, unique constraints, and required fields were implemented exactly according to the ERD.

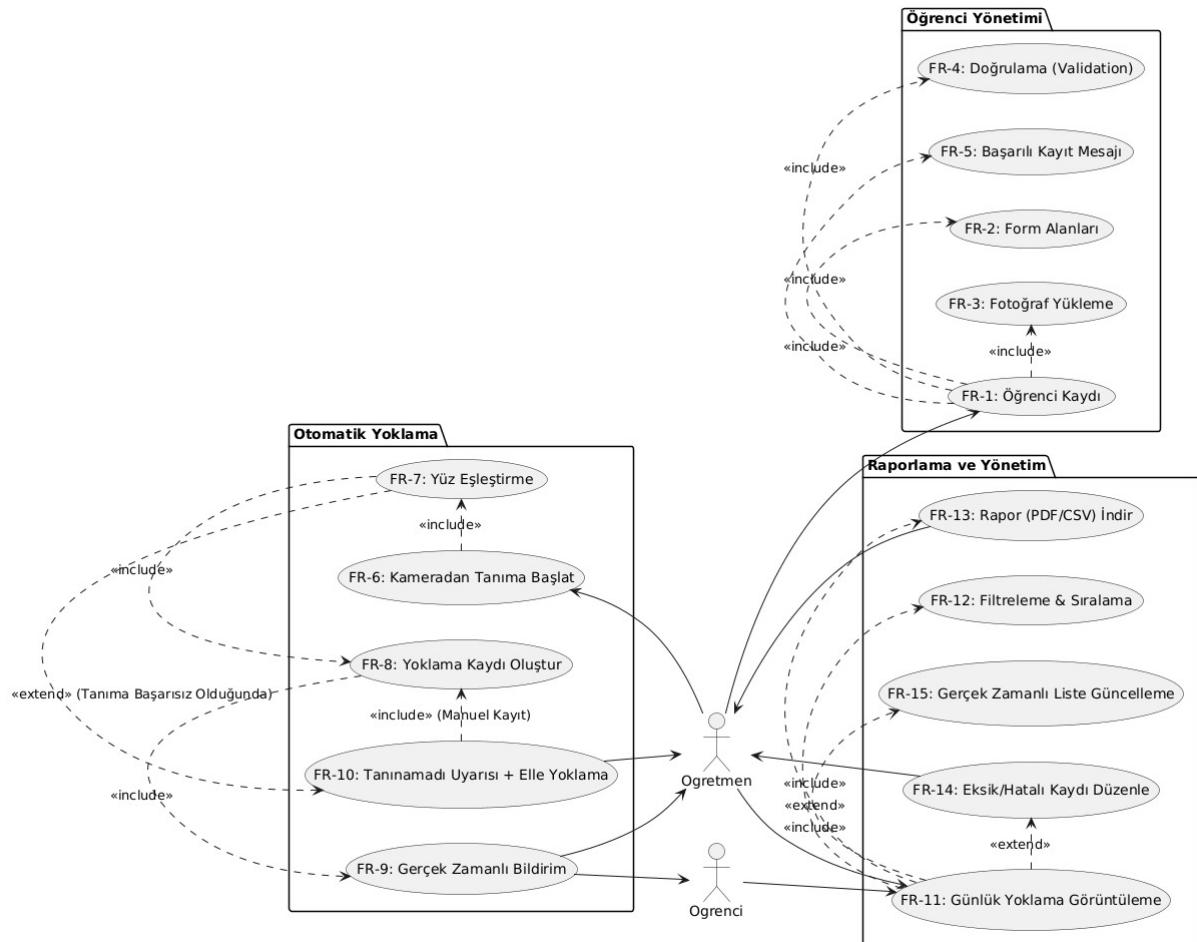
The structure was tested using SELECT queries, and all relationships functioned correctly.



3.3 UML Diagrams

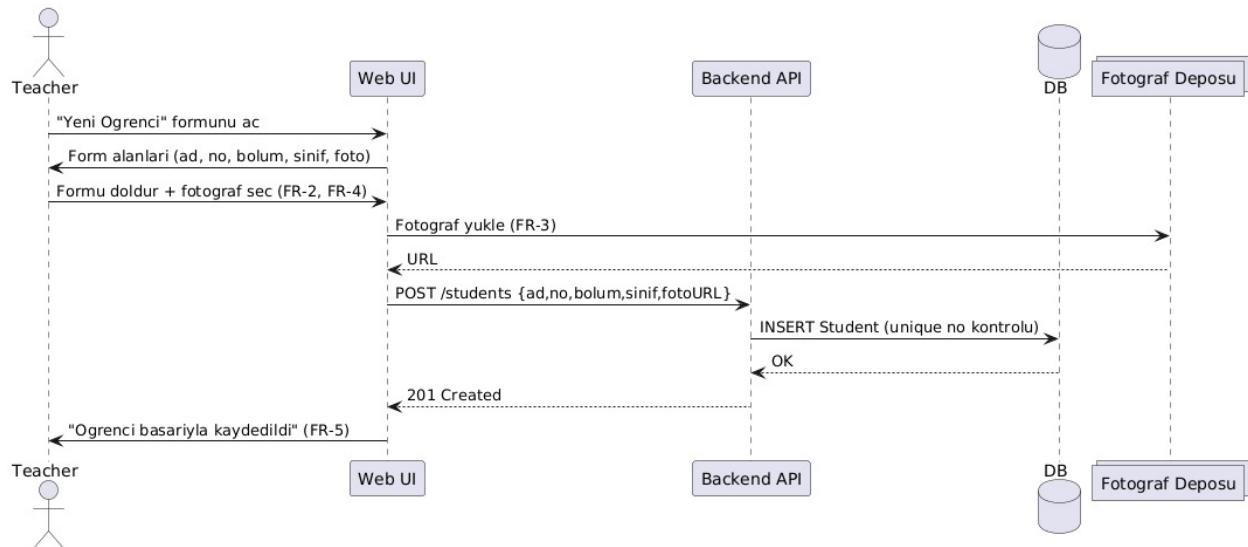
3.3.1 Use Case Diagram

Attendance System Use Case Diagram

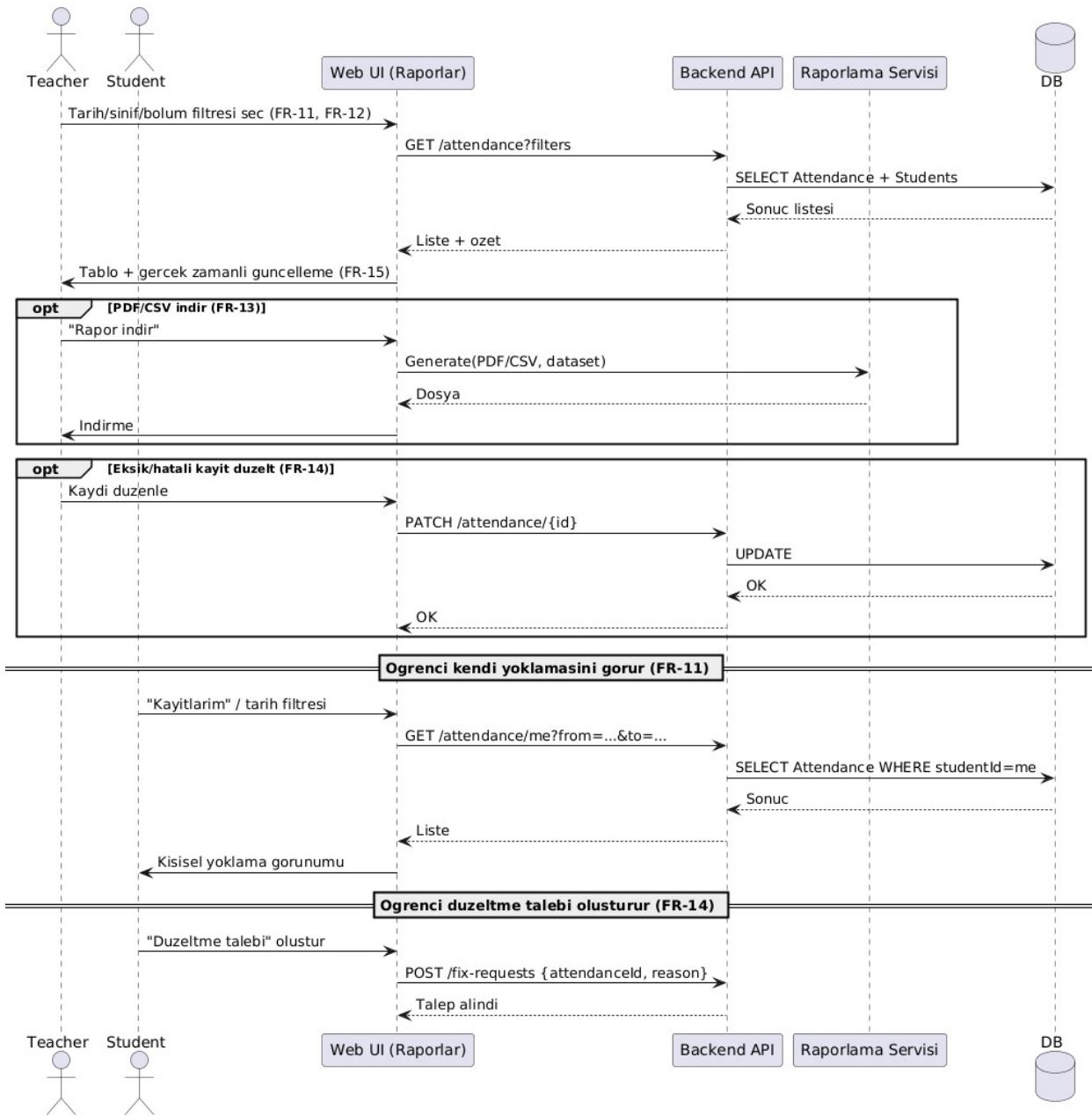


3.3.2 Sequence Diagrams

- Student Registration



■ Automatic Attendance

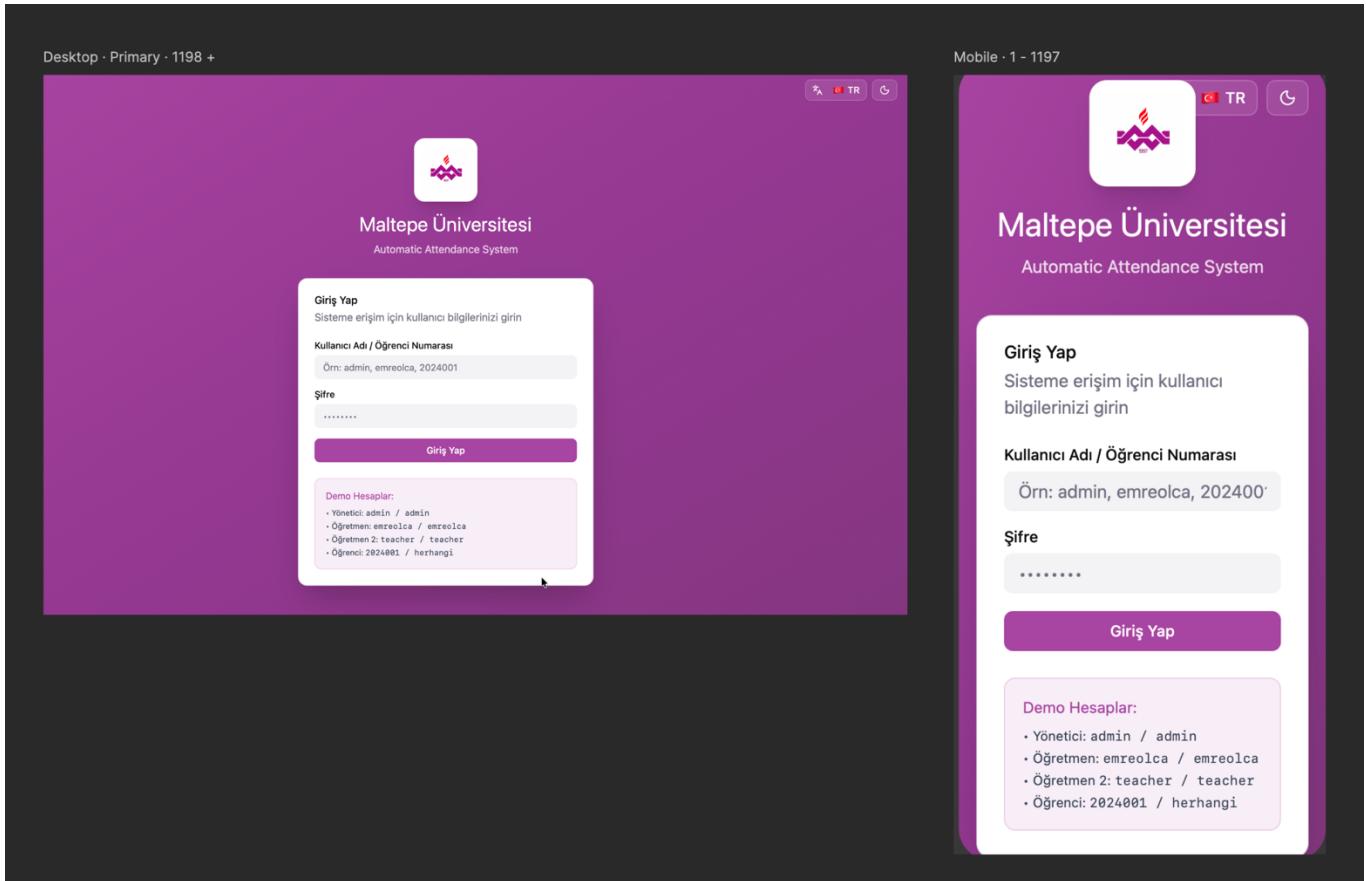


■ Attendance Reporting



4. User Interface (UI) Design

- *Log in Page*



▪ Student Registration Screen

Desktop · Primary · 1198 +

Mobile · 1 - 1197

■ Automatic Attendance Session

The image shows two side-by-side screenshots of an "Automatic Attendance System" interface, one for Desktop (Primary · 1198+) and one for Mobile (1 · 1197).

Desktop Interface:

- Header:** Maltepe Üniversitesi Automatic Attendance System.
- Sidebar:** Includes "Öğrenci Kaydi" (Student Record) with a person icon, "Canlı Yoklama" (Live Attendance) with a camera icon, and "Raporlar" (Reports) with a document icon.
- Main Content:**
 - Canlı Yoklama:** Title "Canlı Yoklama" with subtext "Yüz tanıma sistemi ile otomatik yoklama alın".
 - Ders Bilgileri:** "Ders seçin" dropdown menu with "Yoklamayı Başlat" (Start Attendance) button.
 - Anlık Yoklama Listesi:** "Tanınan öğrenciler (FR-9)" (Recognized students (FR-9)) list with a checkmark icon and the message "Henüz yoklama alınmadı" (Attendance has not been taken yet).
 - Kamera Görüntüsü:** A large video feed placeholder with a camera icon and the text "Kamera devre dışı" (Camera off) and "Yoklamayı başlatın" (Start attendance).
- Footer:** Includes language selection buttons (TR, EN, FR) and a "Çıkış Yap" (Logout) button.

Mobile Interface:

- Header:** Maltepe Üniversitesi.
- Main Content:**
 - Canlı Yoklama:** Title "Canlı Yoklama" with subtext "Yüz tanıma sistemi ile otomatik yoklama alın".
 - Ders Bilgileri:** "Ders seçin" dropdown menu with "Yoklamayı Başlat" (Start Attendance) button.
 - Kamera Görüntüsü:** A large video feed placeholder with a camera icon and the text "Kamera devre dışı" (Camera off) and "Yoklamayı başlatın" (Start attendance).

■ Attendance Reports for Teacher

Desktop - Primary - 1198 +

The desktop application interface includes a sidebar with icons for student records, live attendance, and reports. The main area shows a search bar for student names or numbers, and a table of attendance records. The table columns are Student, Number, Course, Date, Time, Status, Grade, and Action. The records show various student names, course codes like CSE101 and CSE201, and attendance status (Present, Absent, Manual). Buttons for CSV and PDF download are at the top right of the table.

Öğrenci	Numara	Ders	Tarih	Saat	Durum	Güven	İşlem
Ahmet Yılmaz	2024001	CSE101 - Algoritma ve Programlama I	16.11.2024	09:00	Present	95%	View
Ayşe Demir	2024002	CSE101 - Algoritma ve Programlama I	16.11.2024	09:01	Present	92%	View
Mehmet Kaya	2024003	CSE102 - Veri Yapıları	16.11.2024	10:30	Manual	-	View
Zeynep Çelik	2024004	CSE101 - Algoritma ve Programlama I	16.11.2024	09:02	Present	88%	View
Can Öztürk	2024005	CSE201 - Nesne Yönelimli Programlama	15.11.2024	14:00	Present	97%	View
Ahmet Yılmaz	2024001	CSE102 - Veri Yapıları	15.11.2024	10:30	Present	94%	View
Ayşe Demir	2024002	CSE201 - Nesne Yönelimli Programlama	15.11.2024	14:01	Manual	-	View

Höş geldiniz,
Dr. Ahmet Yılmaz
Öğretmen

➡️ Çıkış Yap

Mobile - 1 - 1197

The mobile application interface shows a summary message "Toplam 7 kayıt bulundu" (7 results found) and a "Filtreleri Temizle" (Clear filters) button. Below this, it displays a section titled "Yoklama Kayıtları" (Attendance Records) with a sub-section "Gerçek zamanlı güncellenen kayıtlar (FR-15)". It lists the same attendance records as the desktop version, with each record showing the student's name, ID, course, date, time, status, grade, and a "Düzenle" (Edit) button.

Toplam 7 kayıt bulundu

Filtreleri Temizle

Yoklama Kayıtları

Gerçek zamanlı güncellenen kayıtlar (FR-15)

CSV İndir PDF İndir

Ahmet Yılmaz 2024001 CSE101 - Algoritma ve Programlama I 16.11.2024 • 09:00 %95 Düzenle

Ayşe Demir 2024002 CSE101 - Algoritma ve Programlama I 16.11.2024 • 09:01 %92 Düzenle

- *Student Reports Panel*

Desktop · Primary · 1198 +

Yoklama Takip Panelim
Kendi yoklama kayıtlarınızı görüntüleyin (FR-11)

Toplam Ders: 8 | Katıldım: 6 | Katılmadım: 2 | Devam Oranı: %75

Ders Bazlı Devam Durumu
Her ders için devam oranınız

Ders Kodu	Ders Adı	Devam Oranı (%)
CSE101	Algoritma ve Programlama I	100
CSE102	Veri Yapıları	50
CSE201	Nesne Yönelimli Programlama	50
CSE301	Veritabanı Yönetim Sistemleri	100

Yoklama Geçmişim
Tüm ders katılım kayıtlarınız

Ders Filtresi | Tarih Filtresi | ?

Hoş geldiniz,
Öğrenci 2024001
Öğrenci

Çıkış Yap

Mobile · 1 - 1197

Yoklama Takip Panelim
Kendi yoklama kayıtlarınızı görüntüleyin (FR-11)

Toplam Ders: 8 | Katıldım: 6 | Katılmadım: 2 | Devam Oranı: %75

Ders Bazlı Devam Durumu
Her ders için devam oranınız

Ders Kodu	Ders Adı	Devam Oranı (%)
CSE101	Algoritma ve Programlama I	100

5. Project Plan / Methodology

5.1 Sprint 0: Software Requirement and Analysis

- **Purpose:** To determine and document the project's scope, objectives, and functional/non-functional requirements.
- **Tasks:**
 - SCRUM-48: Software Requirement and Analysis
 - SCRUM-49: Gather Requirements from the Client
 - SCRUM-51: Schedule a meeting with the client and define the project scope...
 - SCRUM-52: Document needs and expectations
 - SCRUM-53: Prioritize requirements by categorizing them
 - SCRUM-50: Requirement Analysis and Documentation
 - SCRUM-54: Classify functional and non-functional requirements.
 - SCRUM-55: Create requirements document (SRS)

5.2 Sprint 1: Database Design and Management

- **Purpose:** To design, install, configure the database schema, and set up security and performance settings.
- **Tasks:**
 - SCRUM-28: Database Design and Management
 - SCRUM-29: Database Schema Design
 - SCRUM-30: Create ER Diagram
 - SCRUM-31: Determine Table and Column Types
 - SCRUM-32: Test Relationships (1-N, N-M)
 - SCRUM-33: Database Setup and Configuration
 - SCRUM-34: Select and Install Database Engine
 - SCRUM-35: Connection Configuration (connection string)
 - SCRUM-37: Develop ORM / Database Connection Layer
 - SCRUM-39: Create Model Classes
 - SCRUM-38: Select ORM Structure (SQLAlchemy, Prisma, Django ORM)
 - SCRUM-40: Database Security and Access Management
 - SCRUM-43: Connection Security (SSL Environment)
 - SCRUM-42: Apply Encryption and Access Policies
 - SCRUM-41: Define User Roles
 - SCRUM-44: Database Backup and Performance Management
 - SCRUM-47: Analyze Query Performance
 - SCRUM-46: Determine Indexing Strategies
 - SCRUM-45: Create Automatic Backup Plan
 - SCRUM-36: Perform Access Tests

5.3 Sprint 2: Student Management and Registration Module

- **Purpose:** To develop the necessary forms and backend integrations for administrators to create new student registrations.

- **Tasks:**
 - SCRUM-25: Student Management and Registration Module
 - SCRUM-1: Create Student Registration
 - SCRUM-5: Create student registration form
 - SCRUM-6: Add form fields (name, number, department, class, photo)
 - SCRUM-7: Add validation rules for form fields
 - SCRUM-8: Develop photo upload function
 - SCRUM-10: Store photos in the system
 - SCRUM-11: Display message after successful registration
 - SCRUM-9: File upload backend integration

5.4 Sprint 3: Camera Integration System

- **Purpose:** To integrate the camera system, set up face recognition and matching algorithms to automate attendance recording.
- **Tasks:**
 - SCRUM-26: Camera Integration System
 - SCRUM-2: Automatic Face Recognition and Attendance with Camera
 - SCRUM-4: Add camera integration
 - SCRUM-12: Set up face recognition and matching system
 - SCRUM-13: Comparison algorithm with existing photos
 - SCRUM-14: Determine the match accuracy threshold
 - SCRUM-15: Create attendance record automatically
 - SCRUM-16: Display real-time "Attendance taken" message

5.5 Sprint 4: Reporting and Admin Panel

- **Purpose:** To create an interface for administrators to view, filter, and manage the attendance records taken.
- **Tasks:**
 - SCRUM-27: Reporting and Admin Panel
 - SCRUM-3: Attendance Reporting and Management
 - SCRUM-18: Create daily attendance viewing page
 - SCRUM-19: Add filtering and sorting feature
 - SCRUM-20: Filter by name
 - SCRUM-21: Sort by department/class
 - SCRUM-17: Warning for unrecognized student and manual attendance entry
 - SCRUM-22: Add module to edit incorrect attendance
 - SCRUM-23: Frontend integration for real-time updates