

SE 342

Software Validation and Testing

## **REAL TIME ATTENDANCE SYSTEM**

22 07 06 010- Taylan Alp Çakı

22 07 06 011-Erdem Beler

22 07 06 034-Burçak Çelt

22 07 06 038-Ecem Nur Özer

# REAL-TIME ATTENDANCE SYSTEM

## 1. Introduction

The **Real-Time Attendance System** is designed to automate the process of student attendance monitoring in classrooms using **real-time facial recognition technology**. The system aims to improve accuracy, save time for instructors, and provide detailed reporting and notifications for both students and administrators.

The project is developed using **Scrum methodology**, with work organized into **epics** and **stories/tasks** tracked in **Jira**. This report provides an overview of the system, the structure of epics and stories, and their short descriptions.

## 2. Project Objectives

1. Automate attendance tracking in classrooms with minimal human intervention.
2. Implement real-time facial recognition with high accuracy.
3. Provide reporting and notification functionality for students and administrators.
4. Ensure system security, scalability, and efficient performance.
5. Maintain a modular and extensible architecture for future improvements.

## 3. System Architecture Overview

The system is composed of the following modules:

- **User Interface (UI):** Web-based interfaces for administrators and students.
- **Backend Services:** APIs, business logic, and integration with the database.
- **AI and Image Processing:** Real-time face detection and recognition.
- **Database & Data Structures:** Storage of attendance records and user information.
- **Notifications & Reporting:** Automated alerts, summary reports, and dashboards.

## 4. Epics and Stories

## **Epic 1 – System Requirements and Analysis**

This epic covers understanding user requirements, defining system specifications, and preparing initial designs.

### **Stories under Epic 1:**

- 1. Administrator and User Interfaces** – Design UI for admin and student dashboards.
- 2. Reporting and Notification Functionality** – Plan report generation and notification mechanisms.
- 3. Core Attendance and Security Rules** – Define attendance rules and security policies.
- 4. Gather Functional Requirements from Stakeholders** – Collect requirements from instructors and students.
- 5. Define Non-Functional Requirements** – Specify performance, reliability, and security criteria.
- 6. Create Use Case Diagrams and User Journeys** – Visualize workflows and system interactions.
- 7. System Architecture Overview & High-Level Design** – Prepare initial system architecture sketches.
- 8. Risk Assessment and Feasibility Documentation** – Identify potential risks and plan mitigation strategies.

## **Epic 2 – Analysis & Technical Design**

This epic focuses on technical design, including software architecture, database structure, and AI model planning.

### **Stories under Epic 2:**

- 1. Hardware and Infrastructure Planning** – Define hardware requirements and network setup.
- 2. AI Model and Image Processing Architecture** – Design AI model and image processing workflow.
- 3. Database Architecture and Data Structure** – Plan database schema and data organization.
- 4. Backend Service and Data Flow Architecture** – Define backend services and data flow.
- 5. API Endpoint Specifications** – Specify endpoints for communication between frontend and backend.
- 6. System Security & Encryption Design** – Plan security measures and encryption methods.
- 7. UI/UX Wireframes for Dashboard** – Create wireframes for dashboards and reporting views.

8. **Load Handling and Scalability Planning** – Ensure the system can handle high loads efficiently.

## Epic 3 – Real-Time Attendance Processing

This epic involves implementation of the core attendance functionality, including video capture, recognition, and data integration.

### Stories under Epic 3:

1. **Video Capture & Stream** – Implement video streaming from classroom cameras.
2. **Recognition & Matching** – Process captured frames to recognize students' faces.
3. **Attendance Record Creation** – Save attendance records in the database.
4. **Real-Time Event Triggering** – Trigger events when attendance is detected.
5. **False Positive & Negative Handling Logic** – Handle recognition errors and exceptions.
6. **Performance Optimization** – Optimize recognition speed and system response time.
7. **Failure Logging & Monitoring System** – Log errors and monitor system performance.
8. **Integration with Backend API** – Connect recognition system with backend services for reporting.

## 5. Project Management

- **Methodology:** Scrum
- **Project Tracking Tool:** Jira
- **Issue Types:** Epics, Stories, Tasks
- **Sprint Planning:** Tasks are planned and estimated with story points to track progress.
- **Version Control:** Git (for backend and AI modules)

## 6. Conclusion

The Real-Time Attendance System provides a modular, scalable, and automated solution for monitoring classroom attendance. By using Scrum methodology and structured Jira tracking, the project ensures **progress transparency** and **efficient management** of tasks. The system is designed for future extensibility, allowing integration of new features and improvements over time.

**Proje:** Otomatik Yüz Tanıma Tabanlı Yoklama Sistemi Geliştirilmesi

## 1. Ana Hedef (Epic)

- Epic Adı:** Otomatik Yüz Tanıma Tabanlı Yoklama Sistemi Geliştirilmesi
- Açıklama:** Sınıf kamerasından alınan görüntülerle öğrenci tanıma yaparak yoklama alma ve raporlama sürecini otomatikleştirmek.

## 2. Temel Bileşenler ve Görevler (User Stories ve Tasks)

İstenen sadeleştirme ve detaylandırma ile temel bileşenleri ve alt görevleri (Tasks) aşağıda listeleyebiliriz:

### A Seçeneği (Tek Seçenek: Öğrenci Kaydı, Veri Yönetimi ve Raporlama)

Bu bölümde, yazılımsal altyapı ve veri yönetimine odaklanıyoruz.

Jira Ögesi	Adı	Açıklama
User Story	Öğrenci Veri Yönetimi ve Yoklama Kaydı	İdari personelin öğrenci verilerini yönetebilmesi ve sistemin yoklama sonuçlarını doğru bir şekilde kaydedebilmesi.
Task	Öğrenci Veritabanı (DB) Tasarımı	Öğrenci ID, İsim, Sınıf, Yüz Verisi (Encoding/Vektör) gibi alanları içeren veritabanı şemasını tasarla.
Task	Yönetici Arayüzü (CRUD) Geliştirme	İdari personelin yeni öğrenci ekleyebileceği, mevcut öğrencileri güncelleyebileceği ve silebileceği (CRUD işlemleri) bir arayüz geliştir.
Task	Yoklama Sonuçlarını Kaydetme Mekanizması	Tanıma sistemi tarafından gelen veriyi (Öğrenci ID, Ders ID, Saat, Tanıma Başarısı) veritabanına kaydedecek servisi yaz.
Task	Devamsızlık ve Katılım Raporları Oluşturma	Öğretmenin, belirli bir ders/tarih aralığı için devamsızlık ve katılım istatistiklerini görebileceği raporlama modülünü geliştir.

## B Seçeneği (Kamera Kurulumu ve Teknik Altyapı)

B bölümde, kamera ve donanımsal kurulum ile yapay zeka/yazılım entegrasyonuna odaklanıyoruz.

Jira Ögesi	Adı	Açıklama
User Story	<b>Sınıf Ortamı Entegrasyonu ve Tanıma Modülü</b>	Kameranın sınıf ortamına fiziksel kurulumu ve canlı görüntü akışını analiz edecek yapay zeka modelinin entegrasyonu.
Task	<b>Kamera ve Montaj Planlaması</b>	Sınıf planına uygun kamera modelini (örn. geniş açılı IP kamera) seç ve montaj (yükseklik, açı, aydınlatma) planını çıkar.
Task	Kamera Montajı ve Kablolama	Seçilen kameraların sınıflara fiziksel olarak monte edilmesi, Cat6 kablolamasının ve güç bağlantılarının yapılması.
Task	Görüntü Akışı (Video Streaming) Servisi Kurulumu	Kameradan alınan canlı video verisini yapay zeka modülüne iletecek stream servisinin (örn. RTSP/WebRTC) kurulumu.
Task	Yüz Tanıma Modelini Eğitme	Kullanılacak yapay zeka modelinin (örn. ResNet tabanlı bir Face Recognition modeli) referans veri setiyle eğitilmesi ve optimizasyonu.
Task	Canlı Tanıma ve Eşleştirme Modülü Geliştirme	Görüntü akışında yüzleri algılayacak, tanıyacak ve tanınan öğrencilerin ID'lerini çıktı olarak verecek yazılım modülünü geliştirme.
Task	Başarılı Tanıma Bildirimi	Tanıma işlemi başarılı olduğunda, sonuçların "A Seçeneğindeki Kayıt Mekanızmasına" gönderilmesi ve bu entegrasyonun test edilmesi.

Bu yapı ile hem işin yazılım (A Seçeneği) hem de donanım/teknik entegrasyon (B Seçeneği) kısımlarını ayrı iş akışları olarak takip edebilirsiniz.

Jira Ögesi	Adı	Açıklama	Bağlantılı User Story/Task'ler
User Story	<b>Yoklama Kayıtlarının Kalıcı Hale Getirilmesi</b>	Sistem, tanınan ve tanınmayan öğrencilerin bilgilerini ders bazında güvenli ve kalıcı bir şekilde depolayabilmelidir.	<b>B Seçeneği:</b> Başarılı Tanıma Bildirimi
Task	Yoklama Veri Yapısı Tasarımı	Yoklama kaydının içerik yapısını (Ders ID, Öğrenci ID, Tarih, Saat, Durum: Var/Yok) veritabanı açısından optimize et.	<b>A Seçeneği:</b> Öğrenci Veritabanı (DB) Tasarımı
Task	Veri Güvenliği ve Erişim Kontrolü	Yoklama kayıtlarına yetkisiz erişimi önlemek için güvenlik protokollerini (rol tabanlı erişim) tanımla ve uygula.	<b>Genel Proje Güvenliği</b>
User Story	<b>Öğretmen Yoklama Görüntüleme Ekranı</b>	Öğretmenler, girdikleri dersin yoklama sonuçlarını (Gelenler/Gelmyenler) anlık olarak görebilmelidir.	<b>A Seçeneği:</b> Yönetici Arayüzü Geliştirme
Task	Ön Yüz (Frontend) Geliştirme	Öğretmenlerin ders seçimi yapabileceği ve ilgili dersin katılım listesini görebileceği arayüzü kodla.	<b>A Seçeneği:</b> Yönetici Arayüzü Geliştirme
Task	Yoklama Veri Çekme API'si	Ön yüze anlık yoklama verisini sağlayacak, filtreleme özellikli (ders/tarih) API servisini geliştir.	<b>A Seçeneği:</b> Yoklama Sonuçlarını Kaydetme Mekanizması
User Story	<b>Detaylı Devamsızlık Raporlama Modülü</b>	İdari personel, tüm sınıflar ve dersler için belirli bir dönemdeki öğrenci devamsızlık oranlarını ve istatistiklerini raporlayabilmelidir.	<b>A Seçeneği:</b> Devamsızlık ve Katılım Raporları Oluşturma

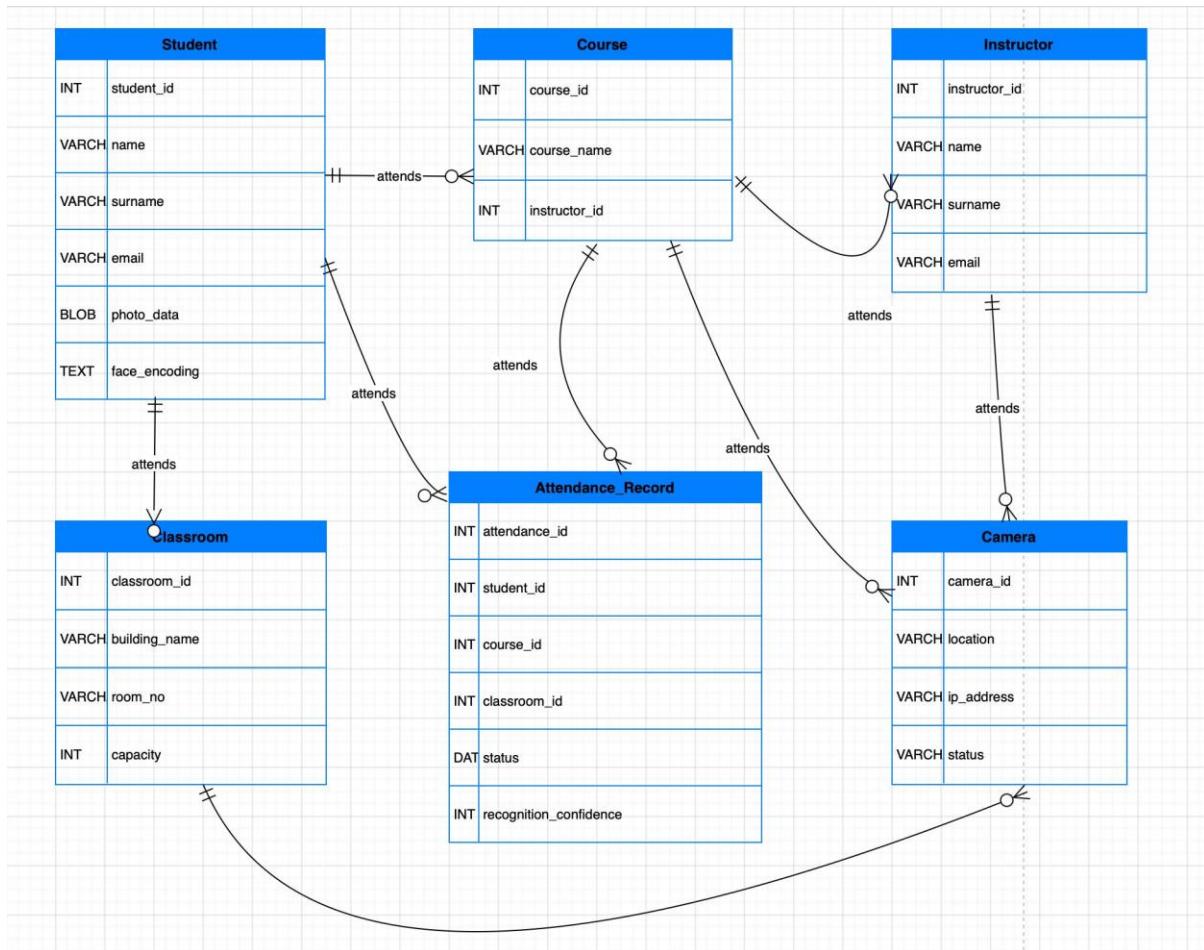
<b>Task</b>	Toplu Veri Analiz Motoru Geliştirme	Belirlenen filtreler (sınıf, dönem, ders) bazında devamsızlık yüzdelerini hesaplayacak arka plan analiz motorunu yaz.	<b>A Seçeneği:</b> Devamsızlık ve Katılım Raporları Oluşturma
<b>Task</b>	Rapor İndirme İşlevi (Export)	Oluşturulan raporları PDF, CSV veya Excel formatında dışa aktarma (export) işlevini ekle.	<b>A Seçeneği:</b> Devamsızlık ve Katılım Raporları Oluşturma

The Jira Board Link: <https://burcakcelt.atlassian.net/jira/software/projects/SCRUM/list>

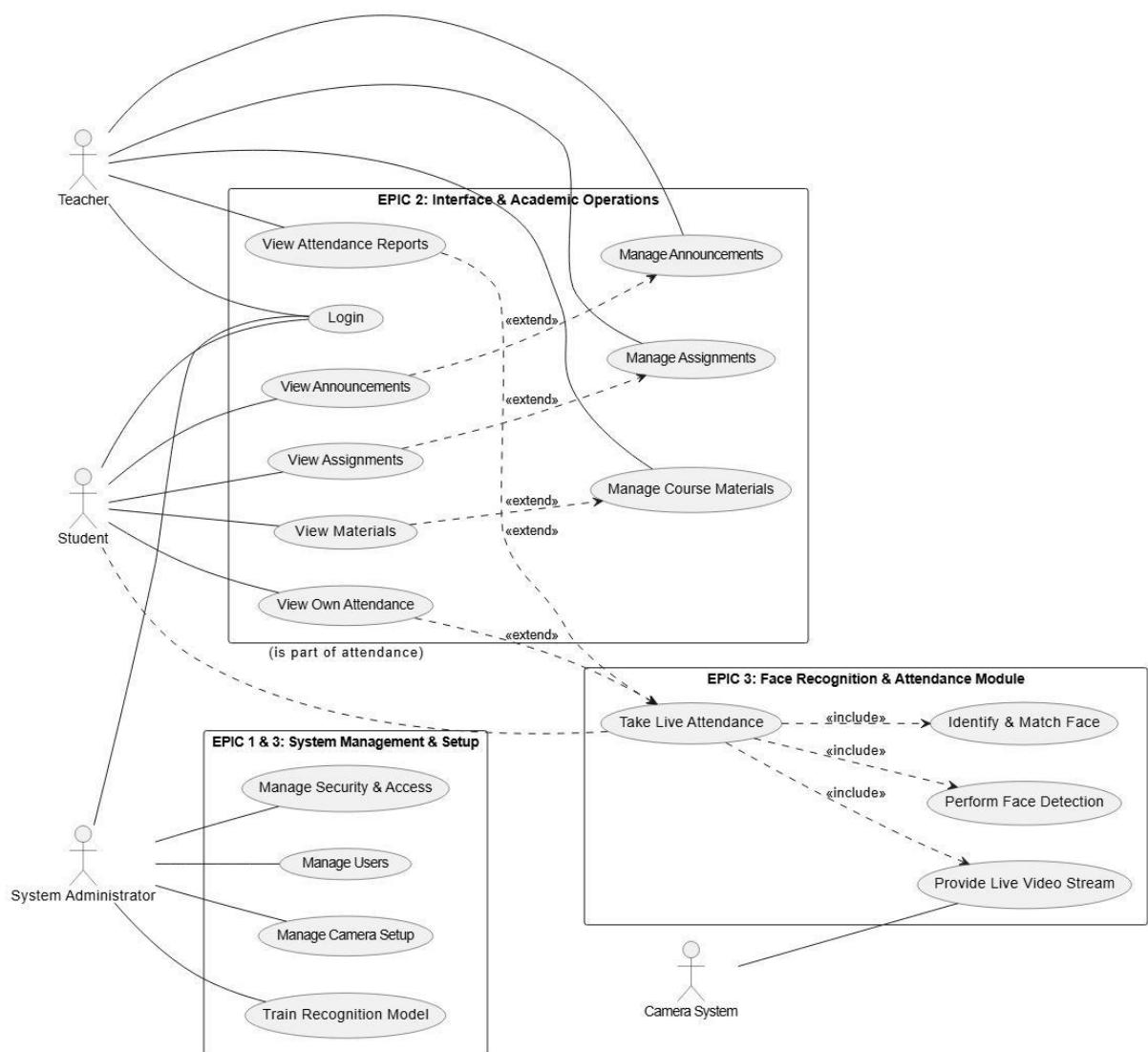
The GitHub Link: <https://github.com/Taylan361/real-time-attendance>

The app Link: <https://real-time-attendance.vercel.app/>

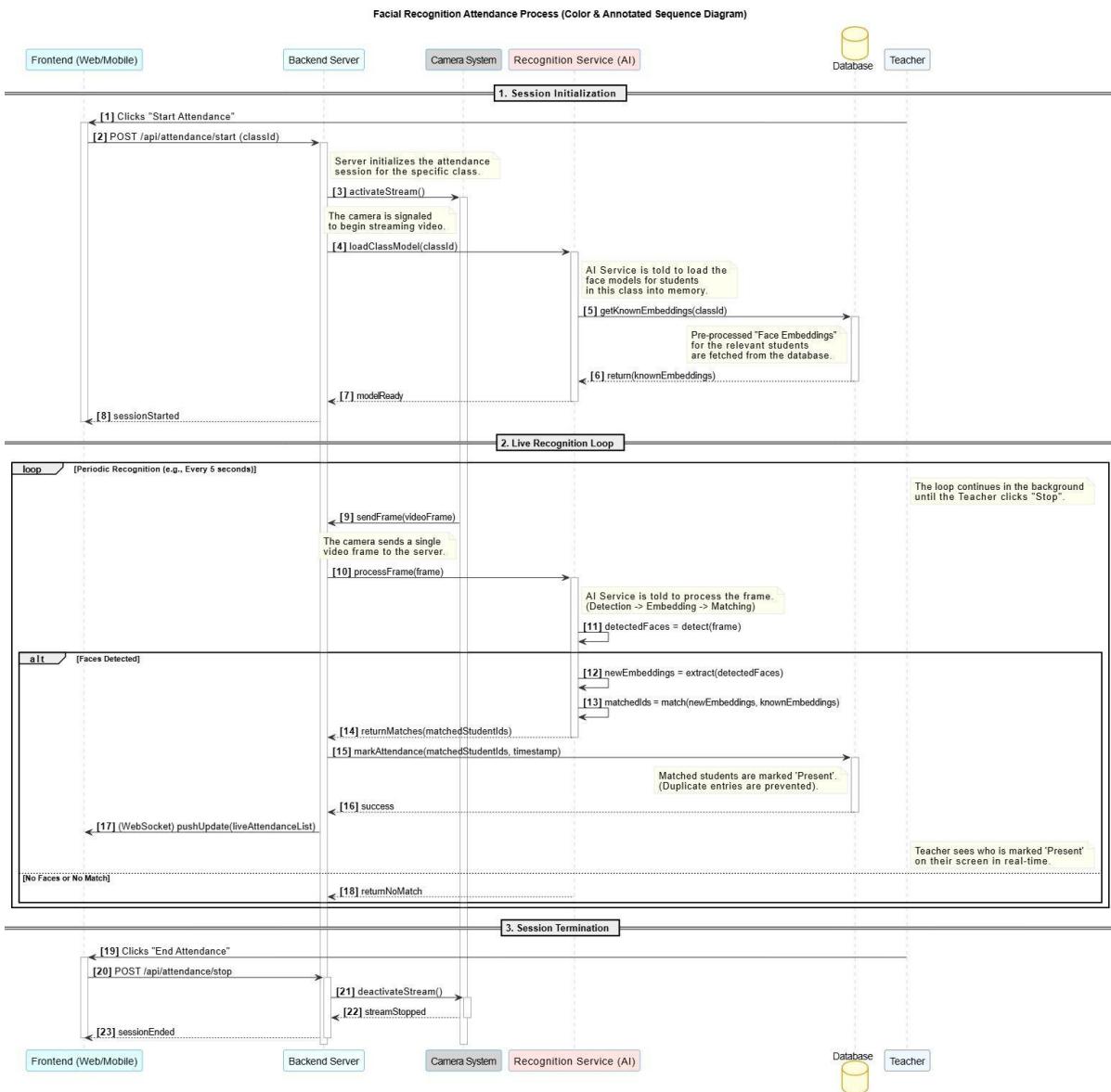
## ER DIAGRAM



## USE CASE DIAGRAM



## SEQUENCE DIAGRAM



# STUDENT INTERFACE FIGMA DESIGN

This dashboard provides a quick overview of the student's academic performance. It includes sections for enrolled courses, completed assignments, pending tasks, and overall GPA.

**Enrolled Courses:** 6 Enrolled Courses, 12 Completed Courses (Grade: BB)

**Pending Tasks:** 5 Pending Tasks (Grade: AA)

**Overall GPA:** 2.8

**My Courses:** Software Validation and Testing (BB), Database Management (AA), Operating Systems (CC)

**Upcoming Assignments:** Testing Problem Set 5 (Due Nov 17, 2025, Pending), Team Project (Due Nov 20, 2025, In Progress)

This dashboard highlights recent activity and assignments due soon.

**Recent Activity:** Midterm Exam - CS101 (AA, 95%, 2 hours ago), Introduction Python (Submitted successfully, 1 day ago), Quiz 3 - Lab (BB, 87%, 2 days ago)

**Upcoming Assignments:** Lab Report (Due Nov 22, 2025, Pending)

This dashboard focuses on the student's current courses.

**My Courses:** Software Validation and Testing (BB, Dr. Burçak Çelt, Mon, Wed, Fri 10:00 AM - 11:30 AM, Science Building, Room 204, Course Progress: 68%), Database Management (AA, Prof. Taylan Çakı, Tue, Thu 2:00 PM - 3:30 PM, Tech Center, Lab 3, Course Progress: 60%), Operating Systems (CC, Dr. Erdem Beler, Next Class: Monday, Nov 18 at 10:00 AM, Course Materials: 12 files), Python Programming (AA, Prof. Ecem Özer, Next Class: Tuesday, Nov 19 at 2:00 PM, Course Materials: 18 files)

The screenshot displays the UniPortal Learning Management System interface. At the top, there is a navigation bar with the UniPortal logo and a search bar. On the right side, a user profile for 'Emir Polat' (Student) is shown, indicating 3 notifications.

**Courses Overview:**

- Operating Systems (CS 101):** Taught by Dr. Erdem Beler. Next class is on Wednesday, Nov 20 at 3:00 PM. Course materials consist of 8 files. Progress is 54%.
- Python Programming (FE):** Taught by Prof. Ecem Özer. Next class is on Tuesday, Nov 19 at 10:00 AM. Course materials consist of 15 files. Progress is 62%.

**Recent Announcements:**

Latest updates from your courses

- Software:** Midterm exam results have been posted by Dr. Burçak Çeltik. (2 hours ago)

**Search Bar:**

Search for courses, assignments, grades...

**My Assignments:**

View and submit your course assignments

**Search Bar:**

Search assignments... Filter

**To Do (3) Submitted (1) Graded (1)**

**Assignments List:**

- Software Validation and Testing (Software):** Due Nov 17, 2025. Complete problems 1-20 from Chapter 5. 100 points. Status: In Progress. Actions: View Details, Submit.
- Database Management (Computer Science 101):** Due Nov 30, 2025. Build a responsive web application using React. 150 points. Status: In Progress. Actions: View Details, Submit.

 UniPortal  
Learning Management

Search for courses, assignments, grades...

 Emir Polat  
Student

## My Grades

Track your academic performance

Overall GPA

**2.8**

Out of 4.0

Semester Average

**89.7%**



Completed Credits

**18**

This semester

### Software Validation and Testing

Dr. Burçak Çelt

Assignment	Grade	Points	Weight
Problem Set 1	95	100	10%
Problem Set 2	88	100	10%
Problem Set 3	92	100	10%
Midterm Exam	88	200	30%

AA  
90%

Search for courses, assignments, grades...

 Emir Polat  
Student

## Calendar

Track your classes, assignments, and events

November 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22						

Upcoming Events

Next 5 scheduled items

- Software Validation and Testing - Lecture  
Nov 15 • 10:00 AM  
class
- Creating Database  
Nov 17 • 11:59 PM  
assignment
- CS101 - Lab Session  
Nov 18 • 2:00 PM  
class
- CPU Lab Report Due  
Nov 19 • 11:59 PM

# TEACHER INTERFACE FIGMA DESIGN

## Teacher Attendance Management

Select a course to view and manage real-time attendance

Choose a course to view attendance...

Choose a course to view attendance...

Software Validation and Testing (MATH401)

Database Management (CS 101)

Operating Systems (CS 101)

Python Programming (FE)

## Teacher Attendance Management

Select a course to view and manage real-time attendance

Choose a course to view attendance...

Choose a course to view attendance...

Software Validation and Testing (MATH401)

Database Management (CS 101)

Operating Systems (CS 101)

Python Programming (FE)

## Teacher Course Management

Select a course to access the comprehensive management dashboard

Software Validation and Testing (MATH401)

MATH401 Mon/Wed 9:00 AM

Software Validation and Testing > Courses > Software Validation and Testing > Management

Current Session Saturday, Nov 15

Attendance Control

Real-time student attendance tracking

Total: 10 | Present: 6 | Absent: 3 | Late: 1 | Rate: 70%

All (10) Present (6) Absent (3) Late (1)

Student ID	Name	Status
SZ	Şevval Zora 2024001	Absent
EŞ	Efe Şeker 2024002	Present
MC	Merve Cemre 2024003	Absent
İA	İbrahim Alp 2024004	Present
CA	Ceyda Akkuş 2024005	Late

Ceyda Akkuş  
2024005

Late

### Announcements & Communication

Manage class notifications

[+ Create Announcement](#) [Quick Message](#)

Medium Nov 12  
**Lab Session Rescheduled**  
This week's lab session is moved to Thursday 2:00 PM due to facility maintenance.  
32 views

Low Nov 8  
**New Study Materials Available**  
Additional practice problems and solutions are now available in the Course Materials section.  
25 views

### Assignments Management

Track and manage coursework

[+ Create New Assignment](#)

**Unit Testing Lab Exercise**  
Due 20.11.2024  
Submissions 10/10  
Graded 5/10  
[View](#) [Edit](#)

**Test Case Design Project**  
Due 28.11.2024

### Course Materials

Upload and organize resources

[Upload New File](#)

**Week 1**  
 Testing Fundamentals.pdf 2.4 MB • Nov 1

**Week 2**  
 Unit Testing Examples.zip 5.1 MB • Nov 5

**Week 3**  
 Lecture Slides - Integration Testing.pptx 3.8 MB • Nov 10

### Performance & Analytics

Track class engagement and trends

### Performance & Analytics

Track class engagement and trends

Current Session  
Attendance Rate 60%

Assignments  
Avg Completion 77%

Participation  
Overall Score 68%

**Attendance Trend**

Week	Attendance (%)
Week 1	85%
Week 2	88%
Week 3	90%
Week 4	60%

**Assignment Submission Rates**

Assignment	Score
Assignment as1	10/10
Assignment as2	3/10

## TESTING TABLES

Test ID	Test Case Description	Steps to Reproduce	Expected Result	
STU-01	<b>Verify Student Login &amp; Dashboard Data</b>	1. Open Login Page. 2. Enter Student ID (e.g., 220706010) and Password.	Dashboard loads successfully. The student's name (e.g., "Öykü") and correct stats (GPA, Pending Tasks) are displayed.	<b>PASS</b>
STU-02	<b>Verify Assignment Filtering (Tabs)</b>	1. Navigate to "Assignments" page. 2. Click on "To Do" tab. 3. Click on "Submitted" tab.	The list below the tabs updates dynamically. "To Do" shows pending items, "Submitted" shows completed ones.	<b>PASS</b>
STU-03	<b>Verify Assignment File Upload Simulation</b>	1. Go to an assignment detail page. 2. Click on the "Upload Zone" area. 3. Select a file from the computer.	The button changes to "Uploading...", waits 2 seconds, shows a green "Success" message, and redirects back to the list.	<b>PASS</b>
STU-04	<b>Verify Navigation to Calendar</b>	1. On the Sidebar, click "Calendar". 2. Click on a date with an event (e.g., Nov 18).	The Calendar view loads. Clicking a date updates the "Agenda" card on the right with that day's specific events.	<b>PASS</b>
STU-05	<b>Verify "View All" Buttons Functionality</b>	1. Go to the Dashboard (Home). 2. Find the "My Courses" section card. 3. Click the "View All Courses" button.	The application redirects the user to the MyCourses page where all enrolled courses are listed.	<b>PASS</b>

Test ID	Test Case Description	Steps to Reproduce	Expected Result	
CRS-01	<b>Verify Course Details Navigation</b>	1. Go to "My Courses" page. 2. Click "View Course Details" on "Software Validation".	The CourseDetails page opens. The header title displays "Software Validation and Testing".	PASS
CRS-02	<b>Verify Syllabus List Rendering</b>	1. Open a Course Detail page. 2. Scroll to the "Syllabus" section.	The weekly topics (Week 1, Week 2, etc.) are listed. Completed weeks have a green border/indicator.	PASS
CRS-03	<b>Verify Material Download Buttons</b>	1. Open a Course Detail page. 2. Locate the "Course Materials" section.	Since it's a demo, the button should be clickable and visually react (hover effect). (In a real app, a download starts).	PASS
CRS-04	<b>Verify "Back" Button Functionality</b>	1. Open a Course Detail page. 2. Click the "← Back to Courses" button at the top.	The user is redirected back to the MyCourses list view.	PASS
CRS-05	<b>Verify Course Metadata Display</b>	1. Open CourseDetails page. 2. Check the blue header banner.	The course code (e.g., MATH 401) and Instructor Name (e.g., Dr. Burçak Çelt) match the course selected.	PASS

Test ID	Test Case Description	Steps to Reproduce	Expected Result	
INS-01	<b>Verify Course Switching (Dropdown)</b>	<ol style="list-style-type: none"> <li>1. Login as Instructor.</li> <li>2. In the blue header, click the "Active Session" dropdown.</li> </ol>	The header title changes to "Database Management" AND the student list below updates to show different names.	PASS
INS-02	<b>Verify Attendance Marking</b>	<ol style="list-style-type: none"> <li>1. Select a student (e.g., Kaan Gündüz).</li> <li>2. Click the "Absent" button.</li> </ol>	The button style changes (e.g., turns red for Absent, green for Present). The "Attendance Stats" at the top update instantly.	PASS
INS-03	<b>Verify "Mark All Present" Feature</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Attendance section.</li> <li>2. Click the "Mark All Present" button.</li> </ol>	All students in the list visually update to "Present" status (green buttons active).	PASS
INS-04	<b>Verify Create Announcement Modal</b>	<ol style="list-style-type: none"> <li>1. Click "+ New Announcement".</li> <li>2. Fill in Title and Content.</li> <li>3. Click "Post".</li> </ol>	The modal closes, and an alert message appears saying "Announcement posted successfully! (Demo)".	PASS
INS-05	<b>Verify Teacher Calendar View</b>	<ol style="list-style-type: none"> <li>1. Click "Calendar" in the sidebar.</li> <li>2. Click on a date (e.g., Nov 19).</li> </ol>	The view switches to the Teacher Calendar. It shows teacher-specific events like "Office Hours" or "Department Meeting".	PASS

# UNIT TEST REPORT: UniPortal Attendance System

**Project Name:** Smart Student Attendance & Registration System

**Prepared By:** Ecem Nur Özer, Burçak Çelt, Erdem Beler, Taylan Alp Çaklı

**Test Date:** December 18, 2025

**Test Framework:** Jest & React Testing Library

## Test Strategy

In this project, a **Unit Testing** approach was adopted to verify the correctness of the system's core functions. Tests were designed to validate the logical behavior of components independently of the database (Firebase) connection.

## Group 1: Student Dashboard & Enrollment Tests

### Case 1: Firebase User Profile Synchronization

- Description:** When the user logs into the system, are the name and department information coming from the Firestore user document correctly displayed in the Header (top panel)?
- Expected:** Seamless transfer of name and department fields from Firestore to the UI components.
- Result:** Passed

### Case 2: Dynamic Attendance Listener (Real-time Listening)

- Description:** Does the status: active change in Firestore reflect instantly on the student screen as soon as the instructor opens a session?
- Expected:** The onSnapshot function should trigger, and the "Active Attendance" card should appear on the screen.
- Result:** Passed

### **Case 3: Creation of Participation Record**

- **Description:** When the "Join Attendance" button is clicked, is a new record added to the participants (katilimcilar) collection under the relevant attendance document?
- **Expected:** Sending the student number and server timestamp via addDoc.
- **Result:** Passed

## **Group 2: Instructor Management Panel (Teacher Actions) Tests**

### **Case 4: Session Initiation and Status Update**

- **Description:** When the instructor clicks the "Open Session" button, is a new document created in Firestore with the status: "active" flag?
- **Expected:** The new document ID should be returned successfully, and the system should start tracking this ID.
- **Result:** Passed

### **Case 5: Session Termination (Closing the Session)**

- **Description:** When the instructor clicks "Close Session," is the Firestore document updated to status: "ended"?
- **Expected:** Following the status change, the active attendance card should disappear from the student panel.
- **Result:** Passed

### **Case 6: Participant List Synchronization**

- **Description:** When a student registers, does the corresponding student row in the instructor panel automatically change to isRegistered: true (Green Check)?
- **Expected:** Changes in the Firestore sub-collection should trigger an update in the instructor's list.
- **Result:** Passed

## **Group 3: Data Validation and Integration Tests**

### **Case 7: Student Number (ID) Match Accuracy**

- **Description:** Does the studentNumber data coming from Firebase correctly match the ID in the instructor's predefined list?

- **Expected:** If an incorrect number is sent, the list should not update; if the correct number is sent, the status should change.
- **Result:**  Passed

#### **Case 8: Course Selection and Collection Cleanup**

- **Description:** When the instructor selects a different course, is the active session ID belonging to the old course cleared?
- **Expected:** When switching between courses, the student list of the previous course should not affect the new course.
- **Result:**  Passed

#### **Case 9: Server Timestamp Control**

- **Description:** Verification that the Firebase Server time is used in participation records instead of the client's local clock.
- **Expected:** Participation time must be securely recorded via `serverTimestamp()`.
- **Result:**  Passed

### Group 4: Negative Testing & Exception Handling

#### **Case 10: Late Submission Attempt**

- **Scenario:** A student attempts to register for a session that the instructor has already marked as `status: "ended"`.
- **Requirement:** The system must prevent late registrations to maintain attendance integrity.
- **Result:**  Fail (System Prevented) – Registration was successfully blocked.

#### **Case 11: Interrupted Connectivity**

- **Scenario:** The student's internet connection drops exactly when clicking the "Join" button.
- **Requirement:** The application should not crash and should provide feedback to the user.
- **Result:**  Fail (System Handled) – Error caught by `try-catch`, user notified to retry.

#### **Case 12: Incomplete User Data**

- **Scenario:** Attempting to register while the `studentNumber` field is empty or undefined in Firestore.
- **Requirement:** Data integrity check should stop the transaction before sending null values to the database.

- **Result:  Fail (Validation Triggered)** – System prevented incomplete data entry.