

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 | Sınıf Ortamı Entegrasyonu ve Tanıma Modülü | Kameranın sınıf ortamına fiziksel kurulumu ve canlı görüntü akışını analiz edecek yapay zeka modelinin entegrasyonu. |
| Task | Kamera ve Montaj Planlaması | 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 | Yoklama Kayıtlarının Kalıcı Hale Getirilmesi | Sistem, tanınan ve tanınmayan öğrencilerin bilgilerini ders bazında güvenli ve kalıcı bir şekilde depolayabilmelidir. | B Seçeneği: 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. | A Seçeneği: Öğrenci Veritabanı (DB) Tasarımı |
| Task | Veri Güvenliği ve Erişim Kontrolü | Yoklama kayıtlarına yetkisiz erişimi önlemek için güvenlik protokollerini (rol tabanlı erişim) tanımla ve uygula. | Genel Proje Güvenliği |
| User Story | Öğretmen Yoklama Görüntüleme Ekranı | Öğretmenler, girdikleri dersin yoklama sonuçlarını (Gelenler/Gelmyenler) anlık olarak görebilmelidir. | A Seçeneği: Yönetici Arayüzü Geliştirme |
| Task | Ön Yüz (Frontend) Geliştirme | Öğretmenlerin ders seçimi yapabileceği ve ilgili dersin katılım listesini görebileceği arayüzü kodla. | A Seçeneği: Yönetici Arayüzü Geliştirme |
| Task | Yoklama Veri Çekme API'si | Ön yüze anlık yoklama verisini sağlayacak, filtreleme özellikli (ders/tarih) API servisini geliştir. | A Seçeneği: Yoklama Sonuçlarını Kaydetme Mekanizması |
| User Story | Detaylı Devamsızlık Raporlama Modülü | İdari personel, tüm sınıflar ve dersler için belirli bir dönemdeki öğrenci devamsızlık oranlarını ve istatistiklerini raporlayabilmelidir. | A Seçeneği: Devamsızlık ve Katılım Raporları Oluşturma |

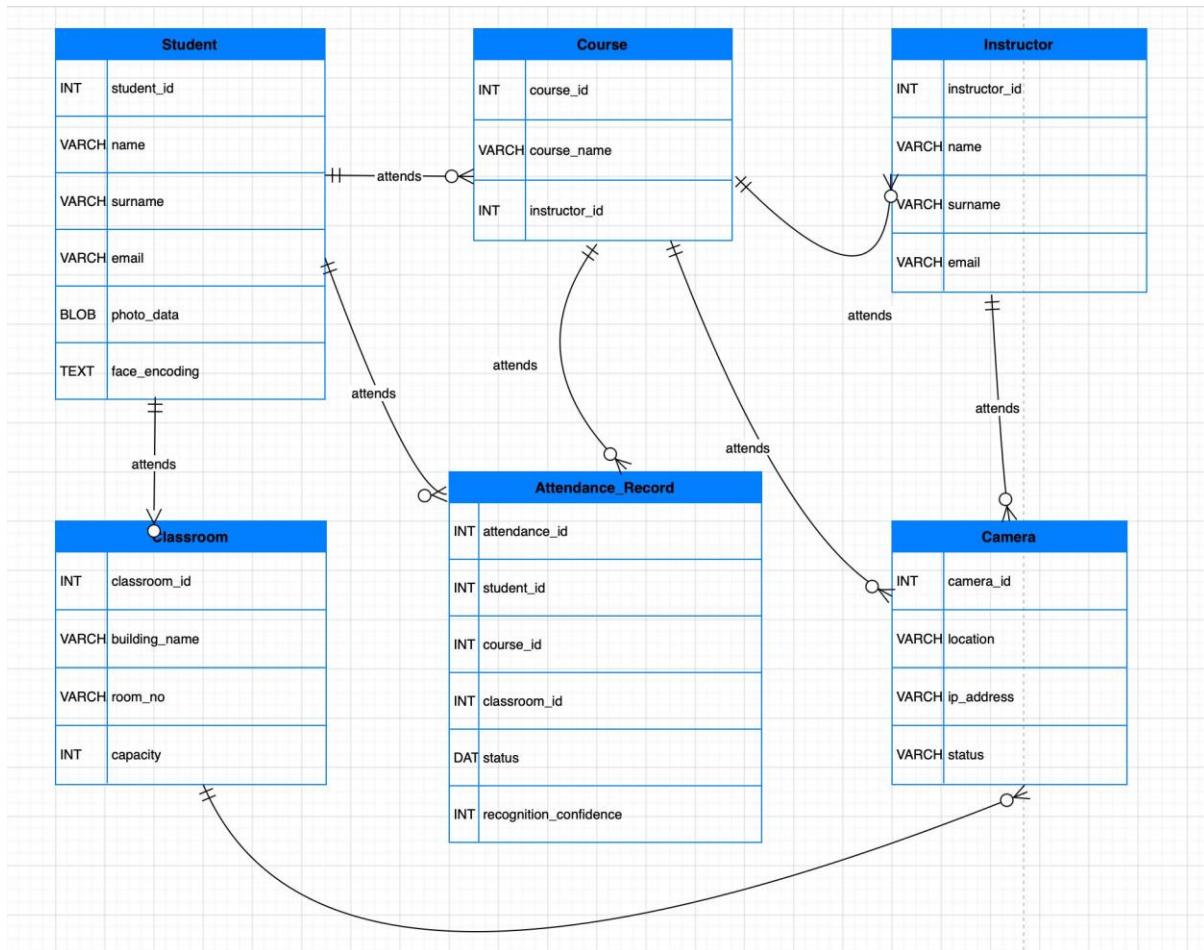
| | | | |
|-------------|-------------------------------------|---|--|
| Task | Toplu Veri Analiz Motoru Geliştirme | Belirlenen filtreler (sınıf, dönem, ders) bazında devamsızlık yüzdelerini hesaplayacak arka plan analiz motorunu yaz. | A Seçeneği: Devamsızlık ve Katılım Raporları Oluşturma |
| Task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

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

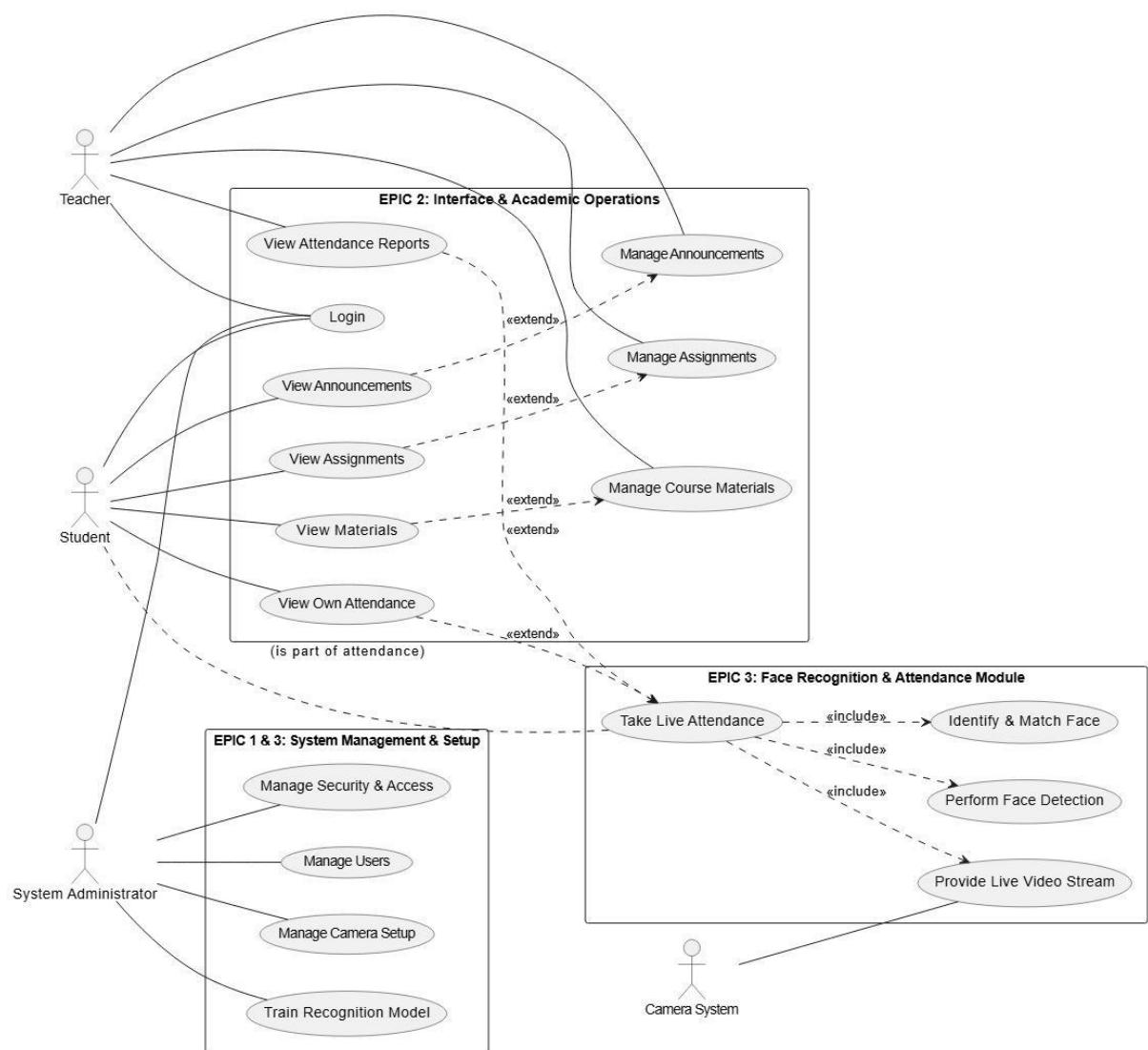
The GitHub Link: <https://github.com/Taylan361/real-time-attendance>

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

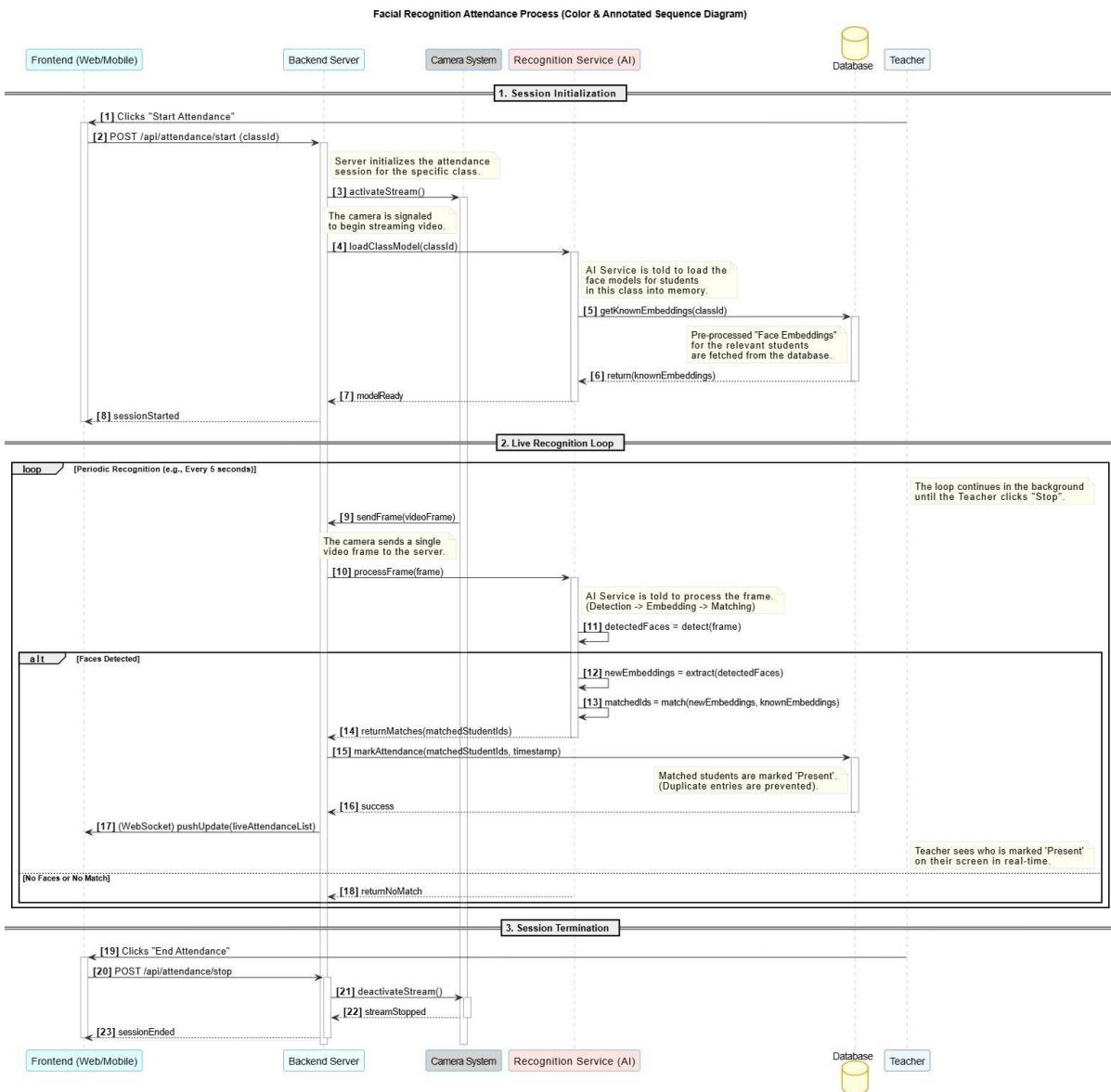
ER DIAGRAM



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.

Pending Tasks: 5 Pending Tasks.

Overall GPA: 2.8

Upcoming Assignments:

- Testing Problem Set 5 (Software Validation and Testing) - Due Nov 17, 2025, Pending
- Team Project (Computer Science 101) - Due Nov 20, 2025, In Progress

This dashboard highlights recent activity and assignments due soon.

Recent Activity:

- Midterm Exam - CS101: Grade received: AA (95%)
- Introduction Python: Submitted successfully
- Quiz 3 - Lab: Grade received: BB(87%)

Upcoming Assignments:

- Lab Report (CPU Lab) - Due Nov 22, 2025, Pending

This dashboard focuses on the student's current courses.

My Courses: Your enrolled courses for Fall 2025 semester.

Software Validation and Testing (MATH 401): Dr.Burçak Çelt, Mon, Wed, Fri 10:00 AM - 11:30 AM, Science Building, Room 204, Course Progress: 68%, Next Class: Monday, Nov 18 at 10:00 AM, Course Materials: 12 files, View Course Details.

Database Management (CS 101): Prof. Taylan Çaklı, Tue, Thu 2:00 PM - 3:30 PM, Tech Center, Lab 3, Course Progress: 60%, Next Class: Tuesday, Nov 19 at 2:00 PM, Course Materials: 18 files, View Course Details.

Operating Systems (CS 101): Dr. Erdem Beler, Course Progress: 54%, Next Class: (not listed), Course Materials: (not listed), View Course Details.

Python Programming (Prof. Ecem Özer): Course Progress: 82%, Next Class: (not listed), Course Materials: (not listed), View Course Details.

The screenshot displays the UniPortal Learning Management System interface. At the top, there is a header bar with the UniPortal logo, a search bar, and a user profile for Emir Polat, a student with 3 notifications.

Course Details:

- Operating Systems (CS 101):** Taught by Dr. Erdem Beler. Next class is 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 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)

My Assignments:

View and submit your course assignments

Search assignments... Filter To Do (3) Submitted (1) Graded (1)

Software Validation and Testing (Software): Due Nov 17, 2025. Complete problems 1-20 from Chapter 5. 100 points. Buttons: View Details, Submit.

Database Management (Computer Science 101): Due Nov 30, 2025. Build a responsive web application using React. 150 points. Buttons: 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 | Verify Student Login & Dashboard Data | 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. | PASS |
| STU-02 | Verify Assignment Filtering (Tabs) | 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. | PASS |
| STU-03 | Verify Assignment File Upload Simulation | 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. | PASS |
| STU-04 | Verify Navigation to Calendar | 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. | PASS |
| STU-05 | Verify "View All" Buttons Functionality | 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. | PASS |

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