SOFTWARE REQUIREMENTS SPECIFICATION

Exam Security System

Major : Software Engineering

Class : Software Validation and Testing

Group Members:

- Deniz Sarıgül 22 07 06 027
- Emre Koç 22 07 06 047
- Yavuz Yaman 22 07 06 042

Contents

1. Introduction

1.1 Purpose

This document specifies the functional and non-functional requirements for the Exam Security System, a web-based application designed to manage exam-day security operations. The system ensures that only registered students enter the exam room, that they sit in assigned seats according to the seating plan, and that all violations are properly recorded and reported.

1.2 Scope

The Exam Security System is a web-based application that supports three core actors:

- Students: Individuals taking the exam
- Proctors (Invigilators): Personnel monitoring exam compliance
- Exam Coordinators (Administrators): Personnel managing exams, seating plans, and reports

The system integrates a simple machine learning/computer vision component for identity verification and provides comprehensive violation logging and reporting capabilities.

1.3 Document Conventions

- Shall/Must: Indicates a mandatory requirement
- Should: Indicates a recommended requirement
- May: Indicates an optional requirement
- FR-X: Functional Requirement identifier
- NFR-X: Non-Functional Requirement identifier

1.4 Intended Audience

- Development Team
- Quality Assurance Team
- Project Stakeholders
- Instructors and Evaluators

2. Overall Description

2.1 Product Perspective

The Exam Security System is a standalone web application that operates independently but may integrate with existing student information systems for roster import. It is designed to be deployed in a controlled exam environment with internet connectivity.

2.2 Product Functions

The system provides the following major functions:

1. Authentication & Authorization: Role-based access control for Proctors and Administrators
2. Exam Management: Create and configure exams with date, time, and room information
3. Seating Plan Management: Define and manage student seating assignments
4. Student Roster Management: Import or manually enter student information
5. Identity Verification: Capture and verify student identity using photo comparison
6. Check-In Workflow: Process student check-in with photo capture and verification
7. Seat Compliance Verification: Validate that students sit in assigned seats
8. Violation Recording: Log and document any exam violations
9. Reporting: Generate reports on check-ins, mismatches, and violations

2.3 User Classes and Characteristics

2.3.1 Students

- Characteristics: Exam participants, may have limited technical experience
- Responsibilities: Provide identity verification, sit in assigned seat
- Frequency of Use: One-time per exam session

2.3.2 Proctors (Invigilators)

- Characteristics: Trained exam monitors, moderate technical experience
- Responsibilities: Verify student identity, check seating compliance, record violations
- Frequency of Use: Throughout exam duration

2.3.3 Exam Coordinators (Administrators)

- Characteristics: Exam management personnel, good technical experience
- Responsibilities: Create exams, manage seating plans, import rosters, generate reports
- Frequency of Use: Before and after exam sessions

2.4 Operating Environment

- Platform: Web-based application (browser-based)
- Browsers: Chrome, Firefox, Safari, Edge (latest versions)
- Server: Node.js/Express or Python/Flask backend
- Database: MySQL, PostgreSQL, or similar relational database
- Hardware: Standard desktop/laptop with camera for photo capture

2.5 Design and Implementation Constraints

- Simple ML/Computer Vision component (library-based, not custom-trained models)
- No deep learning model training required
- Grading focuses on integration, validation, and workflow correctness
- Role-based access control must be enforced
- System must handle concurrent user sessions

2.6 Assumptions and Dependencies

- Students have valid registered accounts with the system
- Photo capture devices (cameras) are available at check-in stations
- Network connectivity is stable during exam sessions
- Database is properly backed up and maintained
- ML/CV library (e.g., face_recognition, OpenCV) is available

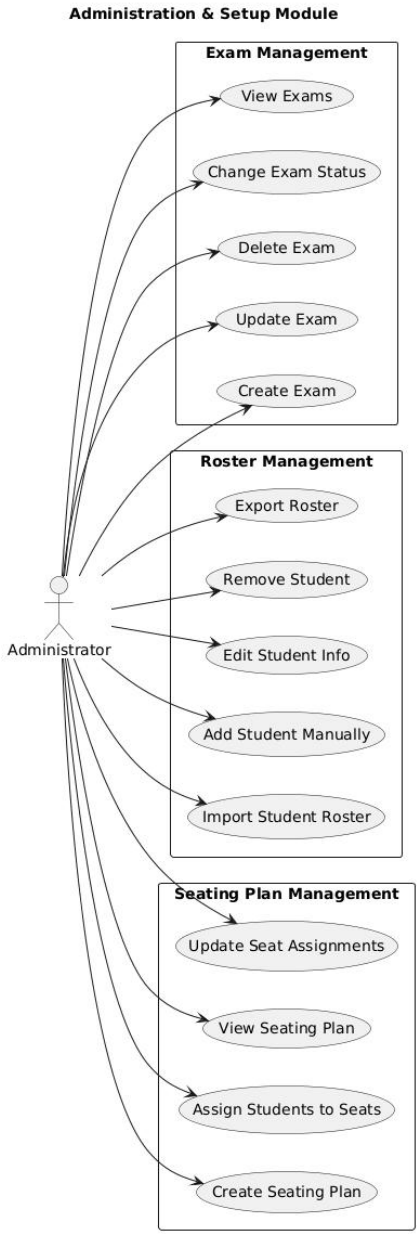
3. System Architecture & Diagrams

This section provides a visual overview of the system architecture through a series of UML diagrams.

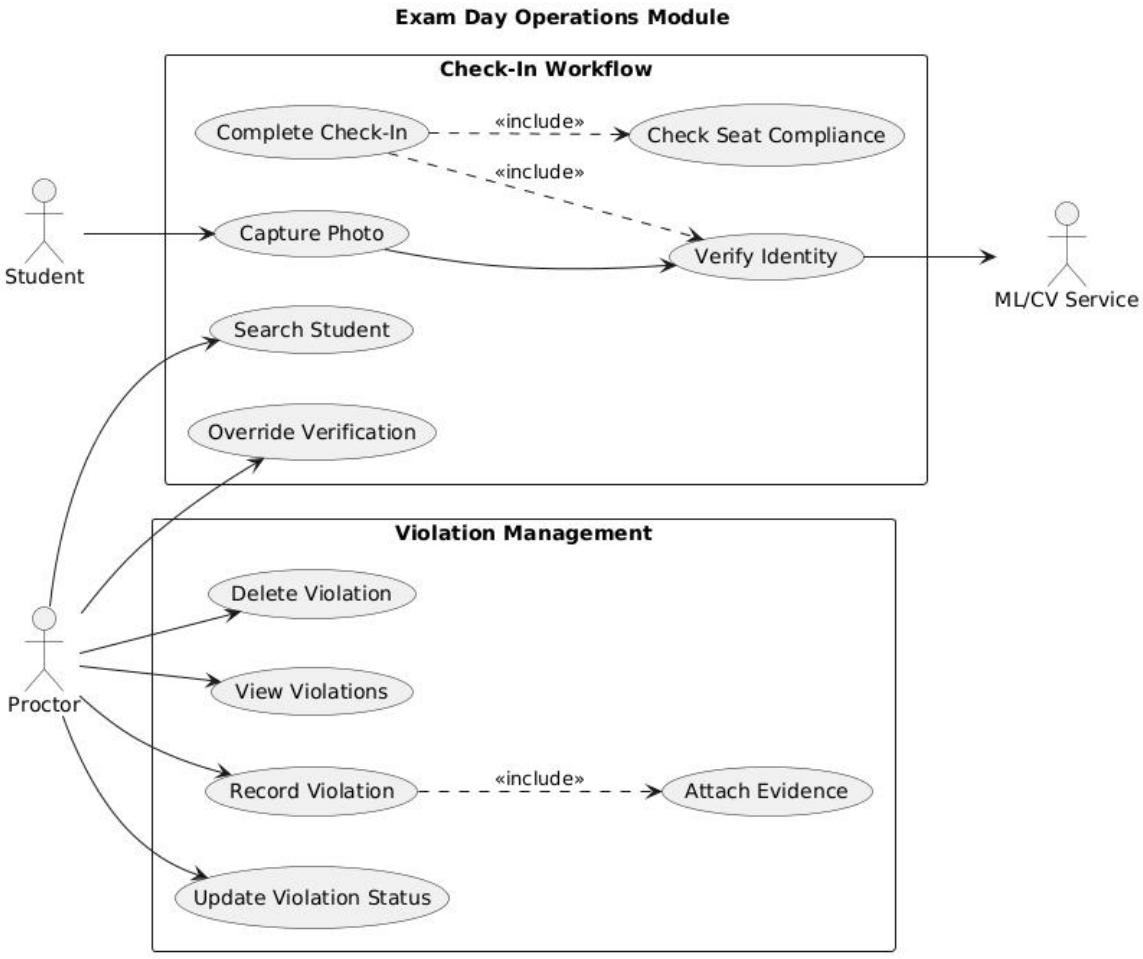
3.1 Use Case Diagram

The Use Case Diagram illustrates the interactions between system actors (Administrator, Proctor, Student, and ML Service) and the main use cases of the Exam Security System. It shows the functional scope of the system from the users' perspective.

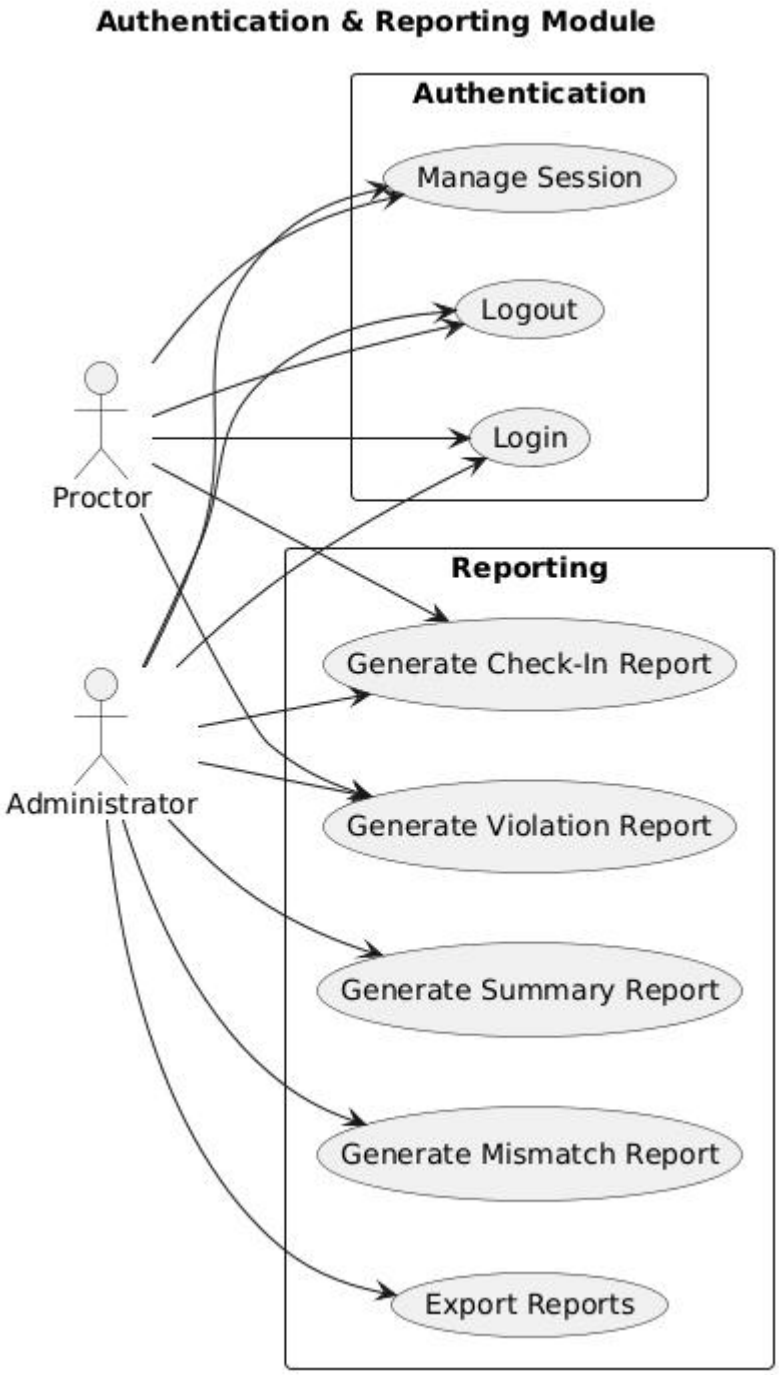
Administration & Setup Module



Exam Day Operations Module

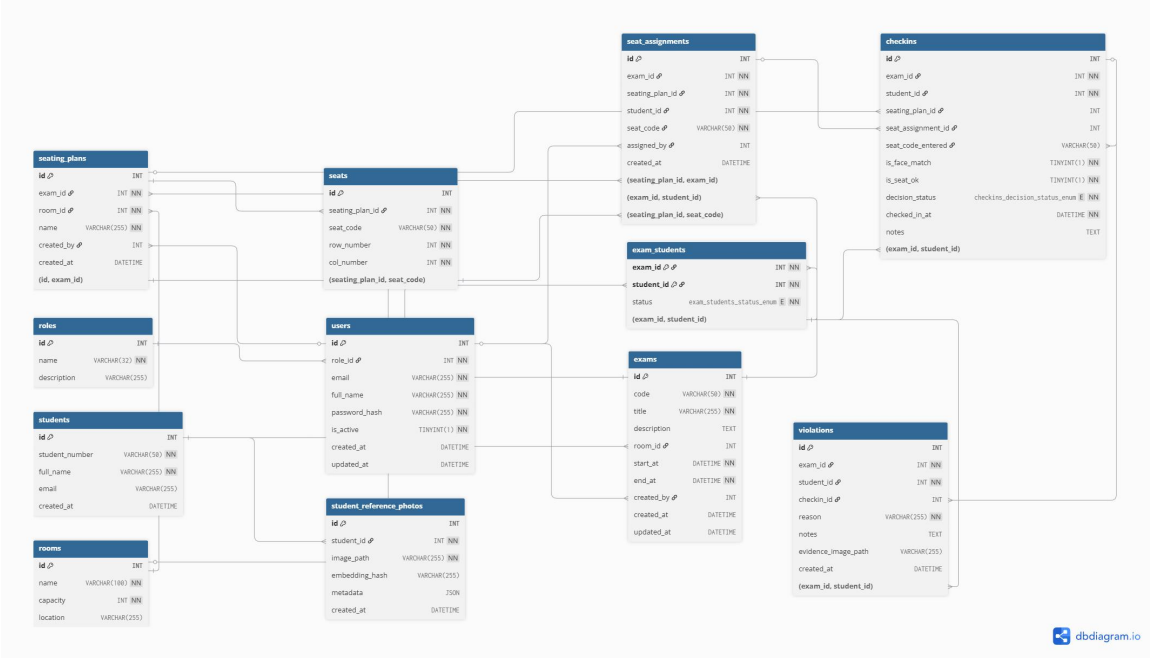


Authentication & Reporting Module



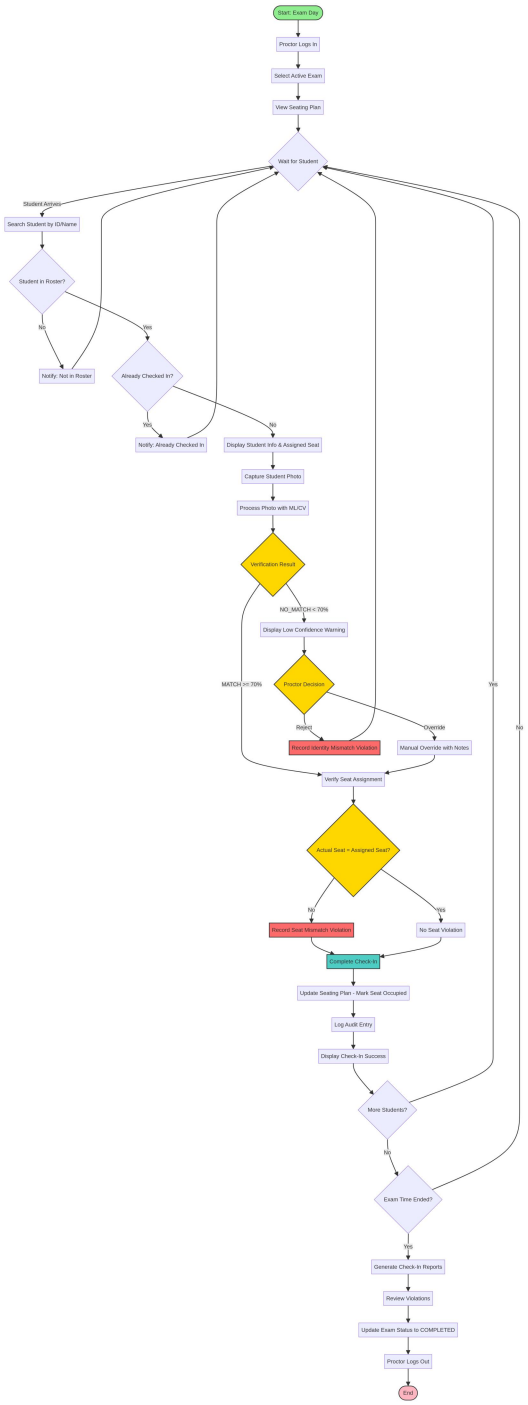
3.2 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram shows the complete database schema, including all entities (tables), their attributes, and the relationships between them. It uses Crow's Foot notation to represent cardinality. The diagram includes 12 main tables: roles, users, students, rooms, exams, seating_plans, seats, exam_students, seat_assignments, student_reference_photos, checkins, violations.



3.3 Activity Diagram

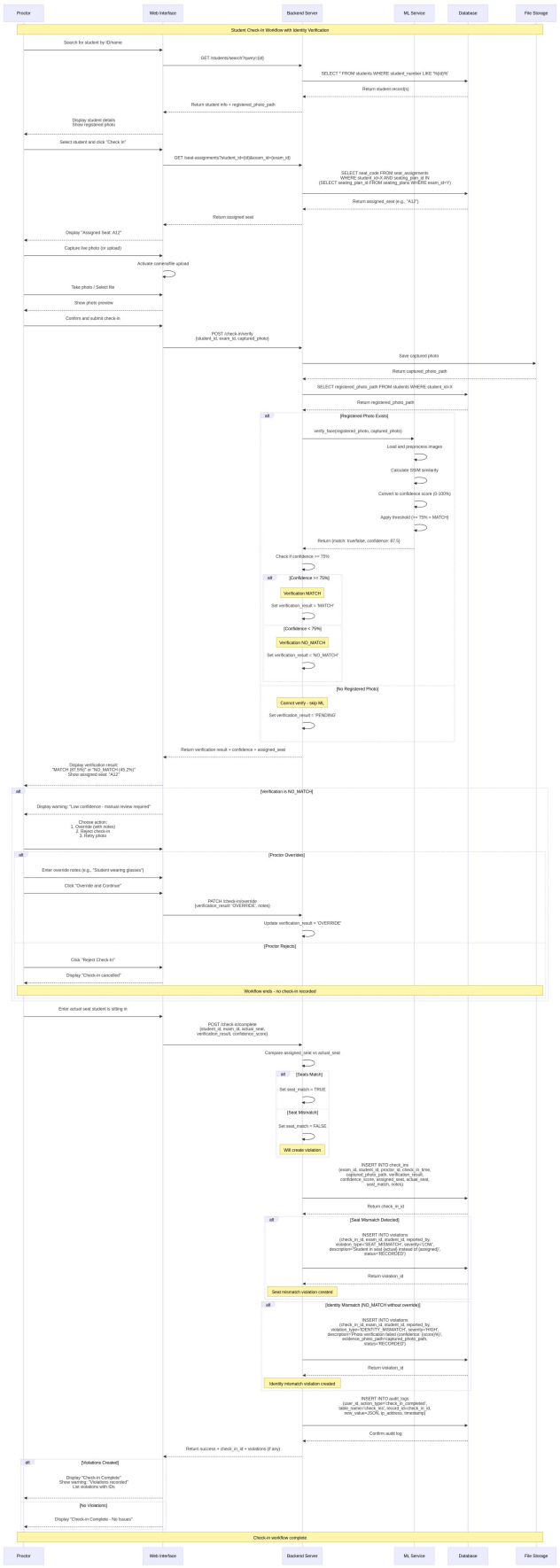
The Activity Diagram provides a high-level overview of the exam day workflow, showing the flow of activities between the Administrator, Proctor, and the System. It illustrates the complete process from exam setup to report generation.



3.4 Sequence Diagrams

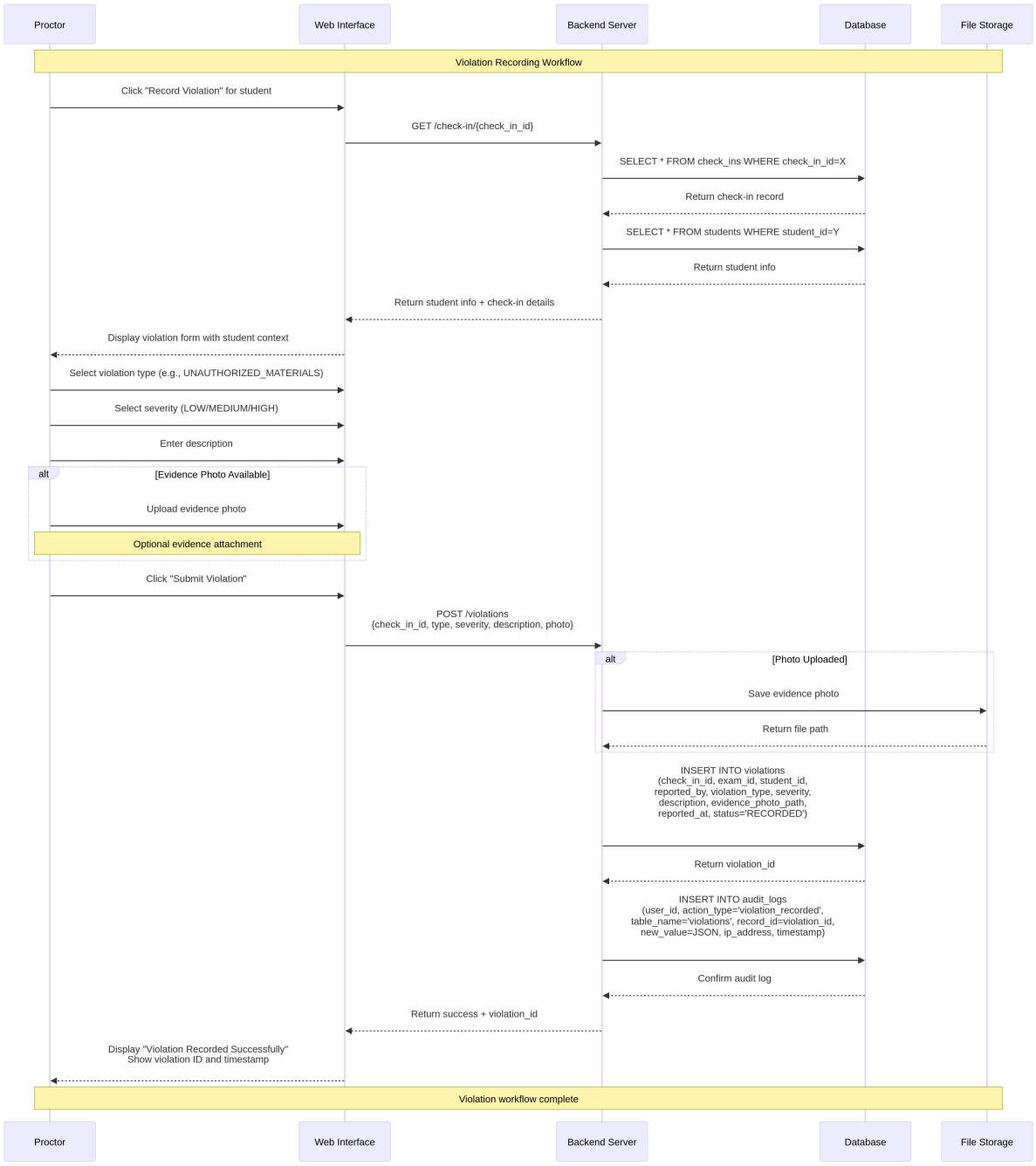
3.4.1 Student Check-In Workflow

This sequence diagram details the step-by-step interactions for the student check-in process, including identity verification using ML/CV, seat compliance checking, and automatic violation creation. It shows the communication between the Proctor, Web Interface, Backend Server, ML Service, Database, and File Storage components.


Proctor
Web Interface
Backend Server
ML Service
Database
File Storage
Student Check-In Workflow with Identity Verification
Search for student by ID/name
GET /students/search?query={id}
SELECT * FROM students WHERE student_number LIKE '%{id}%'
Return student record(s)
Return student info + registered_photo_path
Display student details
Show registered photo
Select student and click "Check In"
GET /seat-assignments?student_id={id}&exam_id={exam_id}
SELECT seat_code FROM seat_assignments
WHERE student_id=X AND seating_plan_id IN
(SELECT seating_plan_id FROM seating_plans WHERE exam_id=Y)
Return assigned_seat (e.g., "A12")
Return assigned seat
Display "Assigned Seat: A12"
Capture live photo (or upload)
Activate camera/file upload
Take photo / Select file
Show photo preview
Confirm and submit check-in
POST /check-in/verify
{student_id, exam_id, captured_photo}
Save captured photo
Return captured_photo_path
SELECT registered_photo_path FROM students WHERE student_id=X
Return registered_photo_path
alt
[Registered Photo Exists]
verify_face(registered_photo, captured_photo)
Load and preprocess images
Calculate SSIM similarity
Convert to confidence score (0-100%)
Apply threshold (>= 75% = MATCH)
Return {match: true/false, confidence: 87.5}
Check if confidence >= 75%
alt
[Confidence >= 75%]
Verification MATCH
Set verification_result = 'MATCH'
[Confidence < 75%]
Verification NO_MATCH
Set verification_result = 'NO_MATCH'
[No Registered Photo]
Cannot verify - skip ML
Set verification_result = 'PENDING'
Return verification result + confidence + assigned_seat
Display verification result:
"MATCH (87.5%)" or "NO_MATCH (45.2%)"
Show assigned seat: "A12"
alt
[Verification is NO_MATCH]
Display warning: "Low confidence - manual review required"
Choose action:
1. Override (with notes)
2. Reject check-in
3. Retry photo
alt
[Proctor Overrides]
Enter override notes (e.g., "Student wearing glasses")
Click "Override and Continue"
PATCH /check-in/override
{verification_result: 'OVERRIDE', notes}
Update verification_result = 'OVERRIDE'
[Proctor Rejects]
Click "Reject Check-In"
Display "Check-In cancelled"
Workflow ends - no check-in recorded
Enter actual seat student is sitting in
POST /check-in/complete
{student_id, exam_id, actual_seat,
verification_result, confidence_score}
Compare assigned_seat vs actual_seat
alt
[Seats Match]
Set seat_match = TRUE
[Seat Mismatch]
Set seat_match = FALSE
Will create violation
INSERT INTO check_ins
(exam_id, student_id, proctor_id, check_in_time,
captured_photo_path, verification_result,
confidence_score, assigned_seat, actual_seat,
seat_match, notes)
Return check_in_id
alt
[Seat Mismatch Detected]
INSERT INTO violations
(check_in_id, exam_id, student_id, reported_by,
violation_type='SEAT_MISMATCH', severity='LOW',
description='Student in seat {actual} instead of {assigned}',
status='RECORDED')
Return violation_id
Seat mismatch violation created
alt
[Identity Mismatch (NO_MATCH without override)]
INSERT INTO violations
(check_in_id, exam_id, student_id, reported_by,
violation_type='IDENTITY_MISMATCH', severity='HIGH',
description='Photo verification failed (confidence: {score}%)',
evidence_photo_path=captured_photo_path,
status='RECORDED')
Return violation_id
Identity mismatch violation created
INSERT INTO audit_logs
(user_id, action_type='check_in_completed',
table_name='check_ins', record_id=check_in_id,
new_value='{SCR, ip_address, timestamp}')
Confirm audit log
Return success + check_in_id + violations (if any)
alt
[Violations Created]
Display "Check-In Complete"
Show warning: "Violations recorded"
List violations with IDs
[No Violations]
Display "Check-In Complete - No Issues"
Check-in workflow complete

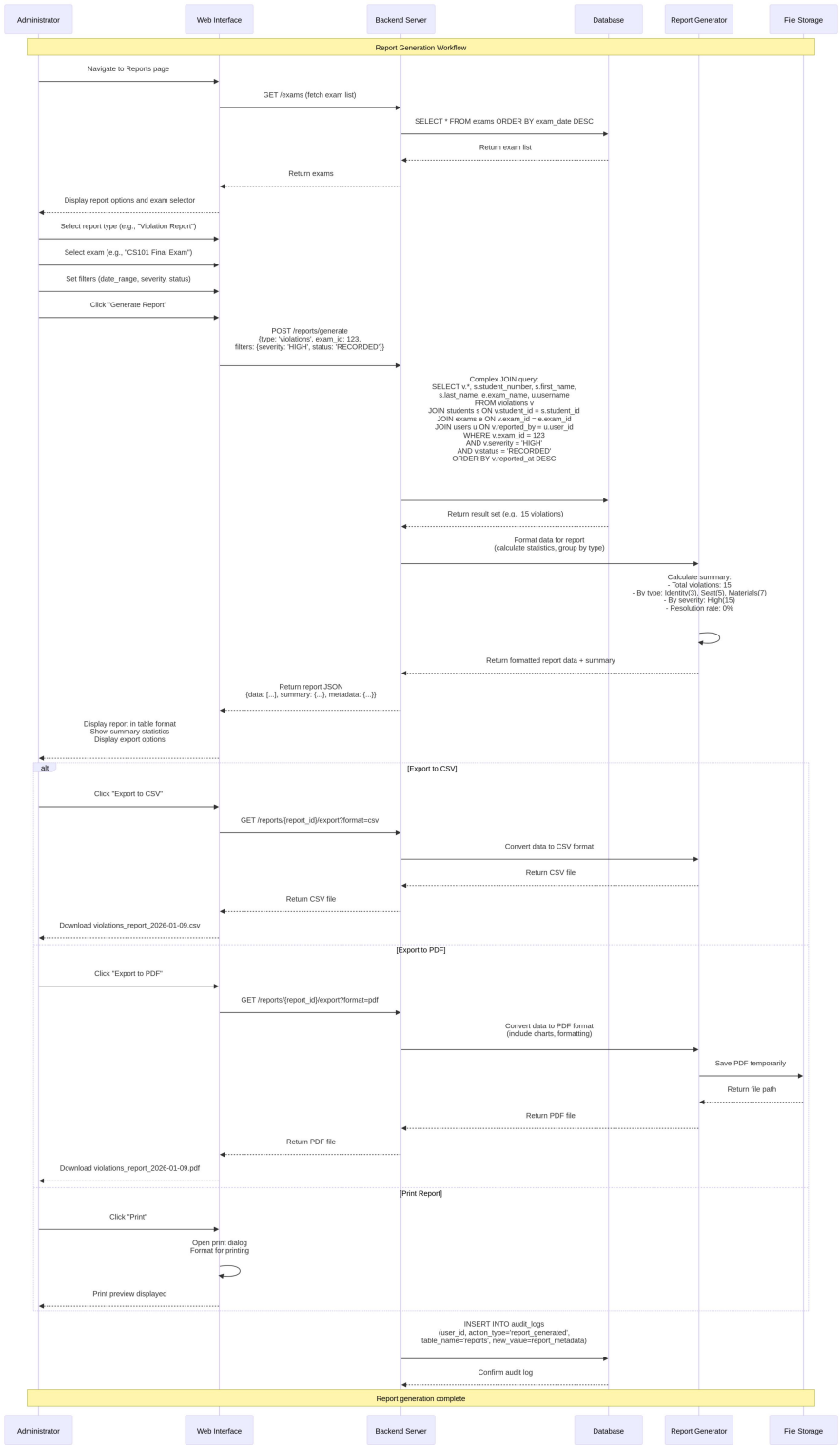
3.4.2 Violation Recording Workflow

This sequence diagram shows the workflow for a proctor recording an exam violation with optional evidence attachment. It includes violation type selection, severity assignment, and audit logging.



3.4.3 Report Generation Workflow

This sequence diagram illustrates the process of an administrator generating and exporting exam reports. It shows the interaction with the Report Generator Service and the various export options (CSV, PDF).



4. Functional Requirements

4.1 Authentication & Authorization

FR-1: User Login

- Description: Users shall authenticate using username and password
- Actors: Proctor, Administrator
- Preconditions: User account exists in the system
- Steps: 1) User navigates to login page, 2) User enters username and password, 3) System validates credentials against database, 4) System creates session and redirects to dashboard
- Postconditions: User is authenticated and session is active
- Alternative Flows: Invalid credentials trigger error message; account lockout after 5 failed attempts

FR-2: Role-Based Access Control

- Description: System shall enforce role-based access control for Proctor and Administrator roles
- Actors: System
- Rules: Proctors can only access check-in, violation recording, and basic reporting features; Administrators can access all features including exam creation, roster management, and advanced reporting; Students do not require login
- Validation: Unauthorized access attempts shall be logged and rejected

FR-3: Session Management

- Description: System shall manage user sessions with automatic timeout
- Timeout Duration: 30 minutes of inactivity
- Actions: Expired sessions redirect users to login page
- Validation: Session tokens are validated on each request

4.2 Exam Management

FR-4: Exam Creation

- Description: Administrators shall create exams with exam name/code, date and time, room/location, duration, and maximum capacity
- Validation Rules: Exam code must be unique, date/time must be in the future, capacity must be greater than 0
- Postconditions: Exam is created and available for roster and seating plan assignment

FR-5: Exam Configuration

- Description: Administrators shall configure exam parameters: enable/disable identity verification requirement, set seating plan requirement (mandatory/optional), configure violation categories and severity levels
- Validation: Configuration changes are logged with timestamp and user ID

FR-6: Exam Status Management

- Description: System shall track exam status: Draft, Active, Completed, Archived
- Transitions: Only authorized transitions are allowed (Draft → Active → Completed → Archived)

4.3 Student Roster Management

FR-7: Student Roster Import

- Description: Administrators shall import student roster via CSV file upload
- File Format: CSV with columns: StudentID, FirstName, LastName, Email, RegistrationNumber
- Validation Rules: StudentID must be unique, required fields must not be empty, duplicate entries are flagged for review
- Postconditions: Students are added to the exam roster

FR-8: Manual Student Entry

- Description: Administrators shall manually add individual students to the roster
- Fields Required: StudentID, FirstName, LastName, Email, RegistrationNumber
- Validation: Same rules as FR-7
- Postconditions: Student is added to roster

FR-9: Roster Management

- Description: Administrators shall manage the student roster: view all students, edit student information, remove students, export roster to CSV
- Validation: Changes are logged with timestamp and user ID

4.4 Seating Plan Management

FR-10: Seating Plan Creation

- Description: Administrators shall create seating plans with Grid-Based (rows and columns) or Seat Code-Based (individual seat codes) options
- Validation Rules: Total seats must be ≥ number of students in roster, seat identifiers must be unique, plan must be associated with an exam
- Postconditions: Seating plan is created and ready for student assignment

FR-11: Student Seat Assignment

- Description: Administrators shall assign students to seats via manual assignment (drag-and-drop or form-based) or automatic assignment (random or sequential)
- Validation Rules: Each student assigned to exactly one seat, each seat assigned to at most one student, cannot assign students not in roster
- Postconditions: All students have assigned seats

FR-12: Seating Plan Visualization

- Description: System shall display seating plan with visual grid representation, student names/IDs in assigned seats, color coding for assigned/unassigned/occupied seats, real-time updates during check-in
- Actors: Proctor, Administrator
- Validation: Display updates within 5 seconds of check-in

4.5 Identity Verification

FR-13: Photo Capture

- Description: System shall capture student photo during check-in
- Process: 1) Student positions face in front of camera, 2) System captures image automatically or on manual trigger, 3) Image is stored with timestamp and student ID
- Technical Requirements: Support multiple image formats (JPEG, PNG), image resolution ≥ 640×480 pixels, automatic face detection to guide student positioning
- Validation: Image must contain a detectable face

FR-14: ML/Computer Vision Verification

- Description: System shall verify captured photo against registered student photo using ML/CV
- Implementation Options: Face verification using face_recognition library, ID photo similarity comparison using OpenCV, basic template matching with embeddings
- Process: 1) Extract face embeddings from captured photo, 2) Compare with registered student photo embeddings, 3) Generate match confidence score (0-100%), 4) Determine pass/fail based on threshold (e.g., 75%)
- Output: Match result (Match/No Match) with confidence score
- Validation: Threshold must be configurable, results must be logged with timestamp, manual override available for Proctors

FR-15: Verification Decision

- Description: System shall present verification decision to Proctor
- Display Information: Captured photo, registered student photo, match confidence score, recommendation (Match/No Match)
- Proctor Actions: Accept verification, reject verification, override decision (manual approval despite mismatch)
- Postconditions: Verification result is recorded with timestamp and Proctor ID

4.6 Check-In Workflow

FR-16: Check-In Initiation

- Description: Proctor initiates check-in process for a student
- Input: Student ID or name search
- Process: 1) System retrieves student information from roster, 2) System displays student details and assigned seat, 3) System prompts for photo capture
- Validation: Student must be in roster and not already checked in

FR-17: Photo Capture & Upload

- Description: System captures and uploads student photo during check-in
- Process: 1) Camera interface is displayed, 2) Student positions face in frame, 3) Photo is captured, 4) Photo is uploaded to server, 5) System confirms successful upload
- Validation: Photo must be valid and contain detectable face

FR-18: ML Verification Decision

- Description: System processes photo through ML/CV component
- Process: 1) System extracts face embeddings, 2) System compares with registered student photo, 3) System generates confidence score, 4) System presents result to Proctor
- Output: Match/No Match with confidence score
- Validation: Result is logged with timestamp

FR-19: Seat Compliance Check

- Description: System verifies that student is sitting in assigned seat
- Process: 1) Proctor confirms student identity, 2) Proctor verifies student is in correct seat, 3) System records seat assignment and check-in time
- Validation: Seat must match student's assigned seat
- Alternative: If student is in wrong seat, violation is recorded

FR-20: Check-In Completion

- Description: System completes check-in process and records result
- Recorded Information: Student ID, check-in timestamp, verification result, assigned seat, actual seat, proctor ID, any violations or notes
- Postconditions: Student is marked as checked-in; seating plan is updated

4.7 Violation Recording

FR-21: Violation Categories

- Description: System shall support violation categories: Identity Mismatch, Seat Mismatch, Unauthorized Materials, Disruptive Behavior, Late Arrival, Other
- Validation: Each violation must have a category

FR-22: Violation Recording

- Description: Proctor shall record violations with violation category, student ID, timestamp, reason/notes, evidence image (optional), severity level (Low/Medium/High)
- Validation Rules: All required fields must be completed, timestamp must be during exam session, evidence image must be valid image file
- Postconditions: Violation is recorded and associated with student

FR-23: Violation Evidence

- Description: Proctor may attach evidence image to violation record
- Process: 1) Proctor captures or uploads image as evidence, 2) System stores image with violation record, 3) Image is linked to violation ID
- Validation: Image must be valid format (JPEG, PNG)

- Optional: Evidence images are optional but recommended

FR-24: Violation Status Tracking

- Description: System shall track violation status: Recorded, Reviewed, Resolved, Dismissed
- Transitions: Only authorized transitions allowed
- Validation: Status changes are logged with timestamp and user ID

4.8 Reporting

FR-25: Check-In Report

- Description: System shall generate check-in report with list of all students checked in, check-in timestamp, verification result, assigned vs. actual seat, proctor who performed check-in
- Format: Exportable to CSV, PDF, or display in web interface
- Filters: By exam, by date range, by proctor

FR-26: Mismatch Report

- Description: System shall generate mismatch report with students with identity mismatches, students in wrong seats, timestamp of mismatch detection, proctor who recorded mismatch
- Format: Exportable to CSV, PDF
- Filters: By exam, by confidence score range

FR-27: Violation Report

- Description: System shall generate violation report with all violations recorded, filterable by category, severity, status, student, proctor, includes violation details and evidence links
- Format: Exportable to CSV, PDF
- Filters: By exam, by date, by violation type

FR-28: Summary Report

- Description: System shall generate summary report with total students checked in, total identity mismatches, total seat mismatches, total violations by category, check-in compliance percentage
- Format: Dashboard view with charts and graphs
- Filters: By exam

5. Non-Functional Requirements

5.1 Performance

- NFR-1: Response Time - All API responses shall be < 1 second under normal load
- NFR-2: Verification Speed - ML/CV verification shall complete in < 3 seconds
- NFR-3: Concurrent Users - System shall support at least 10 concurrent proctors

5.2 Security

- NFR-4: Password Hashing - All user passwords shall be hashed using bcrypt
- NFR-5: Data Encryption - All data in transit shall be encrypted using TLS/SSL
- NFR-6: Audit Trail - All critical actions shall be logged in an immutable audit trail

5.3 Usability

- NFR-7: User Interface - UI shall be intuitive and require minimal training
- NFR-8: Accessibility - System shall comply with WCAG 2.1 AA standards

5.4 Reliability

- NFR-9: Uptime - System shall have 99.9% uptime during exam periods
- NFR-10: Data Integrity - Database shall enforce referential integrity through foreign keys

6. Database Requirements

6.1 Database Schema

The database schema is detailed in the Entity Relationship Diagram (ERD) in Section 3.2. It consists of 12 main tables:

1. **roles**: Defines user roles (e.g., 'admin', 'proctor') to enforce role-based access control.
2. **users**: System operators (Admins and Proctors) with authentication credentials and assigned roles.
3. **students**: Registered student information including student numbers and contact details.
4. **rooms**: Physical exam room information and capacity limits.
5. **exams**: Exam configurations including timing, location, and assigned proctors.
6. **seating_plans**: Layout configurations associated with specific exams and rooms.
7. **seats**: Individual seat definitions (row/column coordinates) belonging to a seating plan.
8. **exam_students**: Enrollment records linking students to specific exams (Roaster).
9. **seat_assignments**: Records linking a specific student to a specific seat for an exam.
10. **student_reference_photos**: Stores paths and metadata for registered student ID photos used for ML verification.
11. **checkins**: Real-time check-in records containing timestamps, ML verification results, and seat compliance status.
12. **violations**: Records of reported anomalies or rule breaches linked to specific students and exams.

7. Business Rules

This section outlines the key business rules that govern the behavior of the Exam Security System. These rules ensure data integrity, security, and consistent operational workflows.

BR-1: Exam Status Progression

- Description: An exam must follow a specific lifecycle: DRAFT → ACTIVE → COMPLETED → ARCHIVED
- Rationale: Ensures that exams are properly configured before becoming active and are properly closed after completion
- Validation: The system will only allow status transitions in the specified order
- Error Handling: An error message will be displayed if an invalid status transition is attempted
- Related Requirements: FR-4, FR-5

BR-2: Proctor-Exam Assignment

- Description: A proctor must be assigned to an exam to perform check-ins and record violations for that exam
- Rationale: Enforces accountability and ensures that only authorized proctors can manage a specific exam session
- Validation: The system will check for a valid proctor-exam assignment before allowing any operational actions
- Error Handling: Access will be denied with a "Not authorized for this exam" message
- Related Requirements: FR-2, FR-15

BR-3: Identity Verification Threshold

- Description: A confidence score of 75% or higher from the ML/CV service is required for an automatic identity match
- Rationale: Balances security with usability by setting a reasonable threshold for automated verification
- Validation: The system will check if confidence_score >= 0.75 to determine the verification result
- Error Handling: Scores below 75% will be flagged as a "NO_MATCH", requiring manual review
- Related Requirements: FR-14, FR-19

BR-4: Account Lockout Policy

- Description: A user account will be temporarily locked after 5 consecutive failed login attempts
- Rationale: Prevents brute-force attacks on user accounts
- Validation: The system will track the number of failed login attempts for each user
- Error Handling: After 5 failed attempts, the user account will be deactivated
- Related Requirements: FR-1

BR-5: Violation-Check-In Association

- Description: All recorded violations must be linked to a valid check-in record
- Rationale: Ensures traceability and context for every violation
- Validation: A foreign key constraint (check_in_id) will enforce this relationship
- Error Handling: The system will prevent the creation of a violation without a valid check-in
- Related Requirements: FR-21

BR-6: Seat Assignment Timing

- Description: Seat assignments for an exam can only be created or modified before the exam start time
- Rationale: Prevents changes to the seating plan while an exam is in progress
- Validation: The system will check if current_time < exam.start_time before allowing modifications
- Error Handling: An error message will be displayed if modification is attempted during active exam
- Related Requirements: FR-11

BR-7: Photo Requirement for Verification

- Description: A registered student photo is mandatory for the ML/CV identity verification process
- Rationale: The system cannot perform a comparison without a reference photo
- Validation: The system will check if the registered_photo_path is not NULL or empty
- Error Handling: If no registered photo exists, manual proctor approval is required
- Related Requirements: FR-13, FR-16

BR-8: Role-Based Report Access

- Description: Proctors can only view reports for exams they are assigned to, while Administrators can view all reports
- Rationale: Enforces data privacy and the principle of least privilege
- Validation: The system will filter report data based on the user's role and exam assignments
- Error Handling: An HTTP 403 Forbidden error will be returned for unauthorized access attempts
- Related Requirements: FR-2, FR-25

8. User Interface Requirements

The following descriptions explain the user interface (UI) screens implemented for the Exam Security System, corresponding to the functional requirements defined in Section 4.

Figure 1: Exam Management

Mapped Requirement: FR-4 (Exam Creation), FR-5 (Configuration)

Actor: Administrator, Proctor (View Only)

Description: This dashboard allows Administrators to view and manage all scheduled exams. The table displays key details including the **Exam Title**, **Start/End Times**, and the assigned **Room**.

Functionality:

Lists all exams sorted by date/time.

Provides links to access detailed views for specific exams.

Proctors use this view to identify their assigned exam sessions.

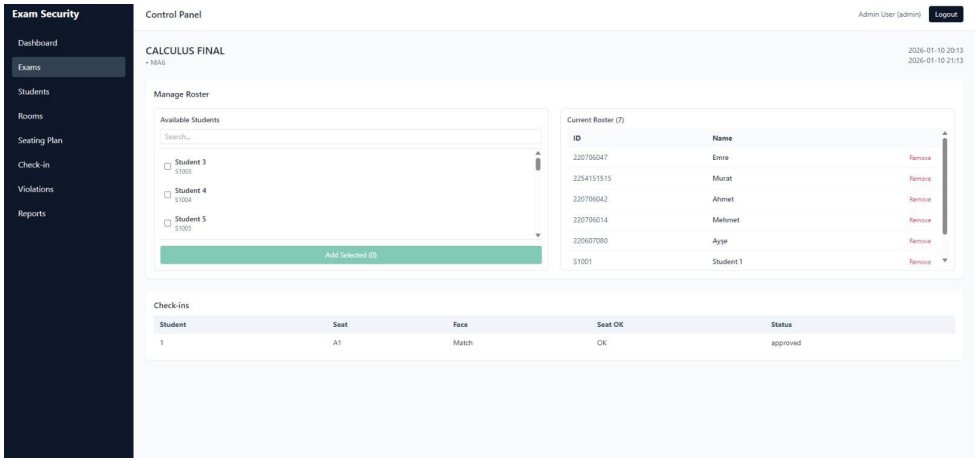


Figure 2: Seating Plan Builder

Mapped Requirement: FR-10 (Seating Plan Creation), FR-11 (Student Seat Assignment)

Actor: Administrator

Description: The Seating Plan Builder provides a visual or manual interface for organizing the exam room layout.

Functionality:

Grid Mode: Admins can define Rows x Columns to automatically generate seat codes (e.g., A1, A2, B1).

Manual Mode: Allows entry of specific seat codes for irregular room layouts.

Assignment: Admins can map specific students from the roster to individual seats, ensuring every student has a designated place before the exam begins.

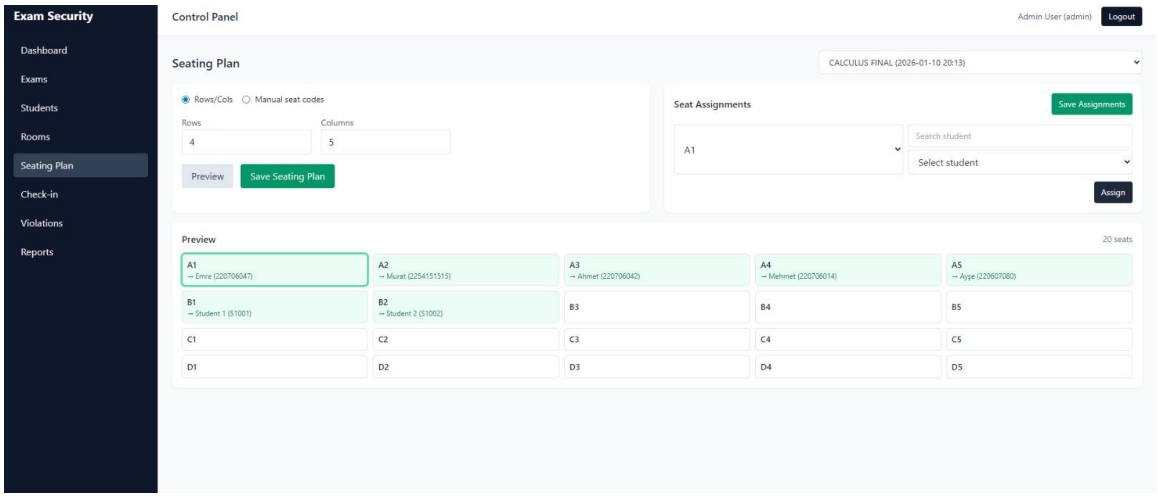


Figure 3: Student Check-In & Verification

Mapped Requirement: FR-16 to FR-20 (Check-In Workflow & Identity Verification)

Actor: Proctor

Description: This is the core operational screen used during exam entry. It integrates the Machine Learning component for identity verification.

Functionality:

Student Selection: The proctor selects the exam and the specific student from the roster.

Photo Capture: Allows uploading or capturing a live photo of the student.

Verification Result: The system immediately returns a "Pass/Fail" decision based on:

Face Match: Comparison between the live photo and the reference photo (via ML).

Seat Compliance: Verification that the entered seat code matches the assigned seat.

Violation Logging: If a mismatch occurs, the proctor can instantly log a violation from this screen.

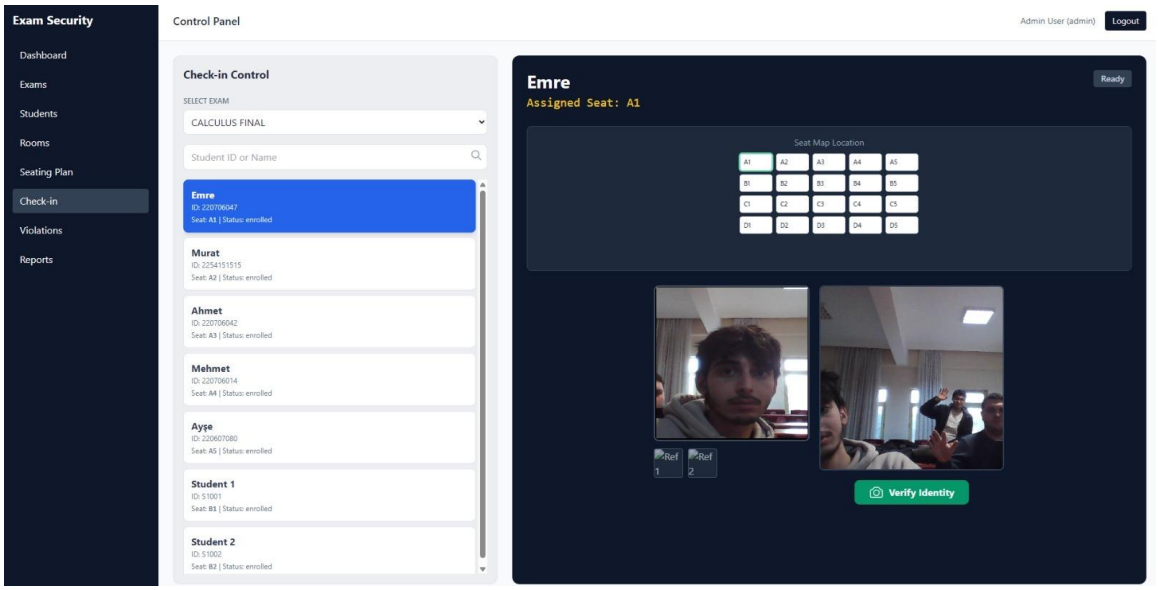


Figure 4: Violation Management

Mapped Requirement: FR-21 to FR-24 (Violation Recording)

Actor: Proctor, Administrator

Description: A centralized log of all anomalies and security breaches recorded during exam sessions.

Functionality:

Displays a tabular view of violations including **Student ID**, **Exam ID**, **Reason** (e.g., Face Mismatch, Wrong Seat), and additional **Notes**.

Allows Administrators to review and audit incidents after the exam.

Ensures traceability of all security events as required by the audit trail policies.

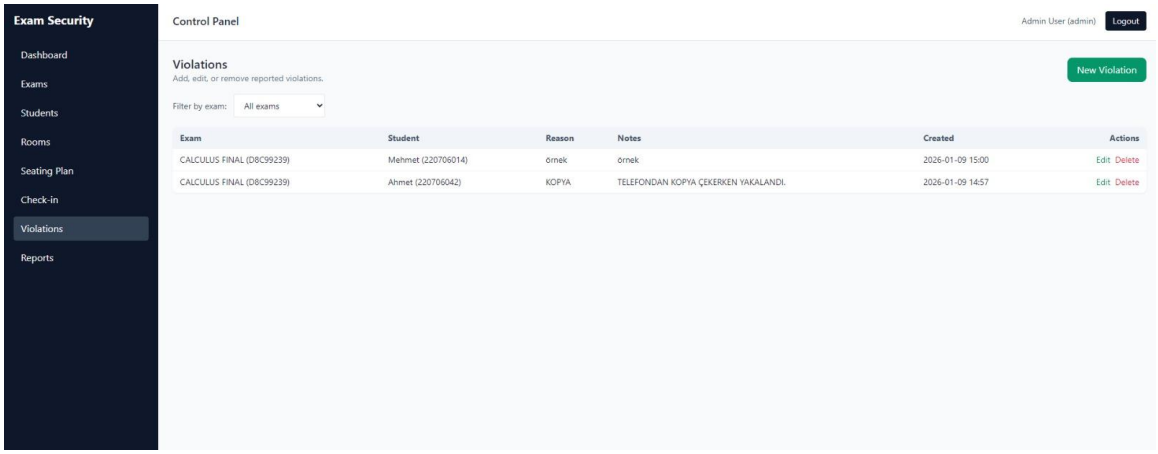


Figure 5: Room Management

Mapped Requirement: FR-4 (Exam Creation - Room Assignment)

Actor: Administrator

Description: This interface manages the physical infrastructure available for exams.

Functionality:

Allows the definition of rooms with specific **Capacity** limits and **Location** details.

Ensures that exams are not assigned to rooms that exceed the student roster size (capacity planning).

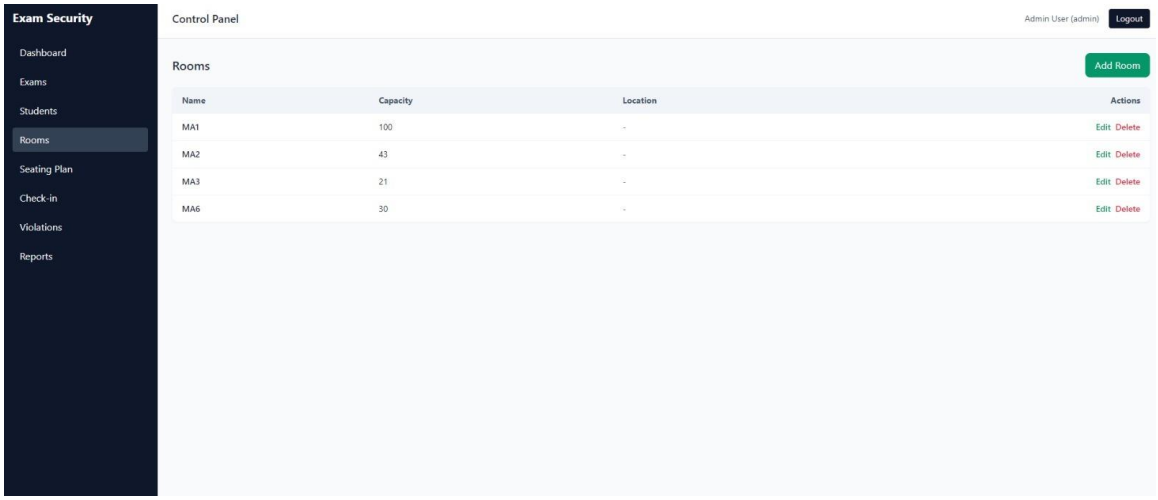


Figure 6: Student Roster Management

Mapped Requirement: FR-7 (Roster Import), FR-9 (Roster Management)

Actor: Administrator

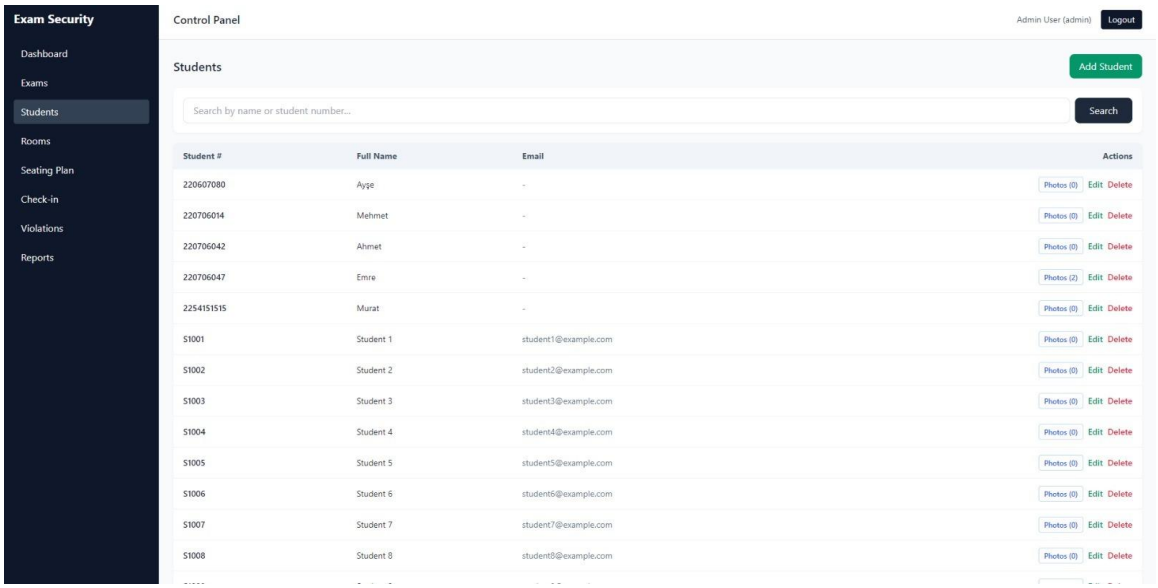
Description: These screens display the registry of students enrolled in the system or specific exams.

Functionality:

Lists student details such as **Student Number**, **Full Name**, and **Email**.

Serves as the source of truth for the Check-In process (only students in this list appear in the Check-In dropdown).

Supports the verification process by linking students to their reference photos.



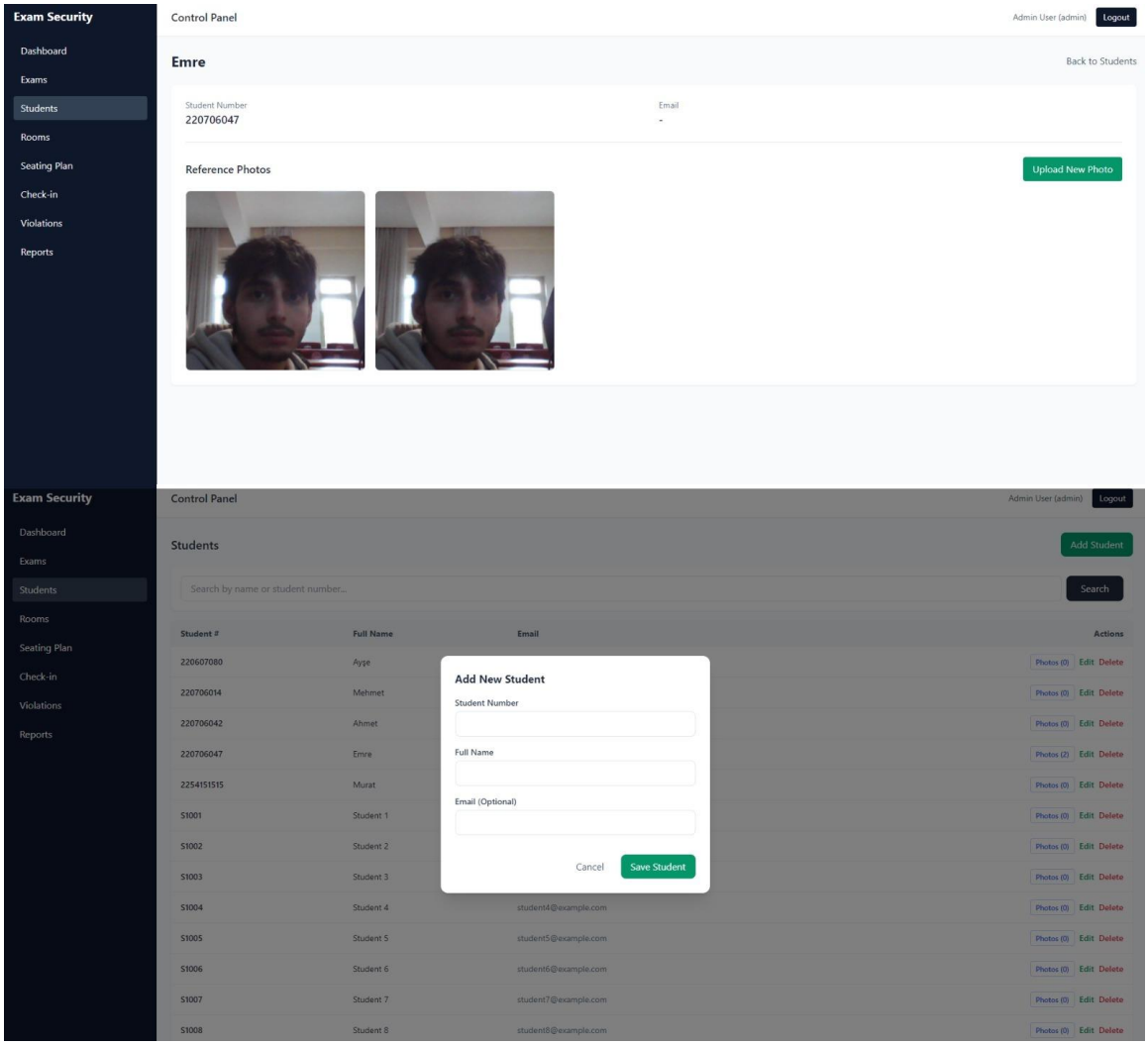

Exam Security
Dashboard
Exams
Students
Rooms
Seating Plan
Check-in
Violations
Reports
Control Panel
Admin User (admin)
Logout
Emre
Back to Students
Student Number
220706047
Email
-
Reference Photos
Upload New Photo
Exam Security
Dashboard
Exams
Students
Rooms
Seating Plan
Check-in
Violations
Reports
Control Panel
Admin User (admin)
Logout
Students
Add Student
Search by name or student number...
Search
Student #
Full Name
Email
Actions
220607080
Ayşe
220706014
Mehmet
220706042
Ahmet
220706047
Emre
2254151515
Murat
S1001
Student 1
S1002
Student 2
S1003
Student 3
S1004
Student 4
student4@example.com
S1005
Student 5
student5@example.com
S1006
Student 6
student6@example.com
S1007
Student 7
student7@example.com
S1008
Student 8
student8@example.com
Photos (0)
Photos (2)
Edit
Delete
Add New Student
Student Number
Full Name
Email (Optional)
Cancel
Save Student

Figure 7: Reports Dashboard

Mapped Requirement: FR-25 (Check-In Report), FR-26 (Mismatch Report), FR-27 (Violation Report), FR-28 (Summary Report)

Actor: Administrator

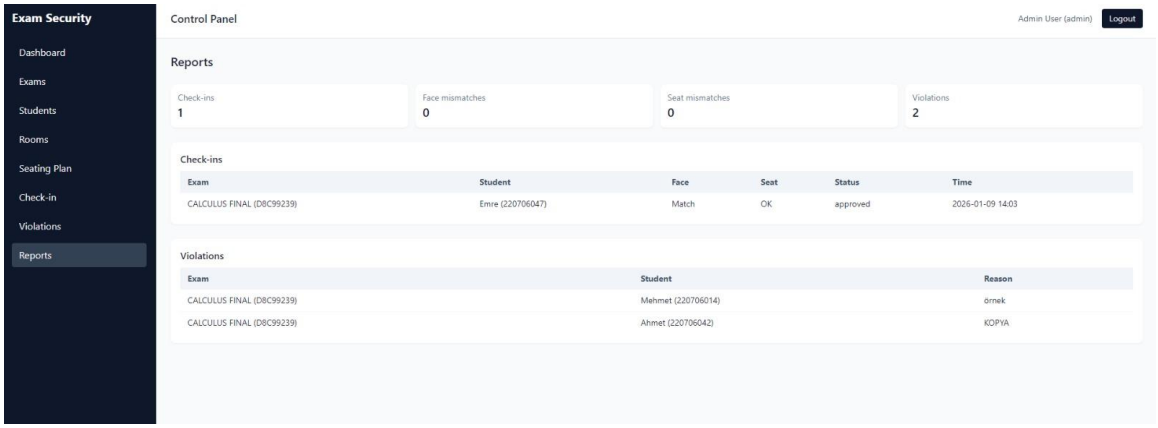
Description: A comprehensive reporting dashboard providing insights into exam security metrics.

Functionality:

Summary Metrics: Displays high-level stats like Total Check-ins, Face Mismatches, Seat Mismatches, and Total Violations.

Check-ins Table: Detailed log of student check-ins, including verification status (Face Match/Mismatch, Seat OK/Wrong).

Violations Table: List of recorded violations for further analysis.



9. Testing & Validation Requirements

The system shall undergo comprehensive testing including:

- Unit Testing: All individual functions and methods
- Integration Testing: API endpoints and database operations
- System Testing: End-to-end workflows
- Performance Testing: Load and stress testing
- Security Testing: Authentication and authorization
- User Acceptance Testing: Real-world scenario validation

Refer to the test-docs/test_cases.md document for detailed test cases.

10. Repository and Project Links

Repository Link:

https://bitbucket.org/examsecuritysystem/exam_security_system/src/main/

Jira:https://yavuzyaman24-1761585477162.atlassian.net/jira/software/projects/ESS/boards/133?atlOrigin=eyJpIjoiMzE1NzI1MWI5MDU1NDM3MjlkOTkxYTdhOTI0ODhmOTUiLCJwIjoiaiJ9