Software Requirements Specification (SRS)

Project: AI-Based Classroom Attendance System

Prepared by: Mohit [Your Team Name]
Date: July 9, 2025
Version: 1.0

1. Introduction

1.1 Purpose

This document defines the software requirements for a Computer Vision-based Attendance System. The system automates the process of taking attendance in classrooms by capturing an image of the entire class, detecting and recognizing student faces, and marking attendance in the database.

1.2 Scope

The system enables:

- Automated attendance based on face recognition
- Storage and retrieval of student attendance records
- Web API for integration with front-end/admin panel
- Admin functionality to register new students and manage records

This solution is intended for educational institutions to reduce manual effort and improve accuracy in attendance tracking.

1.3 Definitions, Acronyms, and Abbreviations

Term	Meaning
CV	Computer Vision
API	Application Programming Interface
DB	Database
ROI	Region of Interest (face bounding box)
YOLO	You Only Look Once (object detection model)
IoU	Intersection over Union

2. Overall Description

2.1 Product Perspective

This is a standalone system that may later be integrated into an existing LMS or ERP. The system consists of:

- Face Detection & Recognition module
- Image preprocessing and embedding module
- Attendance marking backend
- Database system (SQL/NoSQL)
- API layer for frontend interaction

2.2 Product Functions

- Capture classroom image
- Detect faces in the image
- Recognize each face using stored embeddings
- Match recognized students with registered database
- Mark them present for the current date
- Provide APIs to retrieve or edit attendance records
- Admin panel for registration and management

2.3 User Classes and Characteristics

User Role	Description
Admin	Registers new students and manages records
Teacher	Captures image and verifies attendance
Student	Gets marked present if detected

2.4 Operating Environment

- Server: Ubuntu 22.04 LTS or Windows 10
- Languages: Python 3.10+
- Libraries: OpenCV, face_recognition, FastAPI, SQLAlchemy
- Database: PostgreSQL or MongoDB

- Deployment: Dockerized, hosted on cloud/local server

2.5 Design and Implementation Constraints

- High-quality image is required for accurate recognition
- Face dataset needs to be pre-enrolled
- Uniform lighting and student posture enhance performance
- GDPR and data privacy must be considered

3. Specific Requirements

3.1 Functional Requirements

FR-1: Student Registration

- Admin uploads multiple face images of a student
- System computes and stores face embeddings

FR-2: Attendance Capture

- Teacher uploads or captures a classroom photo
- System detects all faces
- Embeddings are compared to the database
- Recognized students are marked “Present”

FR-3: Attendance Record Management

- Attendance is saved with timestamps
- Duplicate entries for the same day are not allowed
- Admin can override status (e.g., mark “Absent”)

FR-4: API Endpoints

- `POST /register_student`: Upload student data and face images
- `POST /mark_attendance`: Submit class photo and auto-mark attendance
- `GET /attendance/{date}`: Get attendance list for a day
- `GET /student/{id}`: View attendance history of a student

3.2 Non-Functional Requirements

ID	Requirement
NFR-1	Response time for detection & recognition must be < 5 seconds
NFR-2	System should handle a class of up to 100 students
NFR-3	Attendance accuracy should be ≥ 95%
NFR-4	System must log unknown or unrecognized faces
NFR-5	All APIs must be authenticated and rate-limited

3.3 Performance Requirements

- Image resolution supported: minimum 720p
- System can process 1 photo at a time (batch option future scope)
- Embedding matching within 0.2 seconds per face (using FAISS)

3.4 Database Design (Brief)

Tables:

students
| id | name | roll_number | embedding | registration_date |

attendance
| id | student_id | date | status (Present/Absent) |

4. Future Scope

- Real-time attendance from CCTV video feed
- Integration with school/college ERP
- Face spoof detection for authentication
- SMS/email notification to absentees
- Graphical dashboard for analytics

5. Appendices

5.1 Tools & Libraries

- **Face Detection:** YOLOv8, OpenCV, MediaPipe
- **Face Recognition:** `face_recognition`, ArcFace
- **Backend:** FastAPI
- **DB:** PostgreSQL with pgvector extension
- **Storage:** AWS S3 or local filesystem
- **Embedding Matching:** FAISS