

## EXPERIMENT 1

---

# **Software Requirements Specification**

**for**

# **Smart Attendance System Using Face Recognition**

**Version 1.0 approved**

**Prepared by Soumyadeep Dey**

**Sharda University**

**30-07-2025**

# Table of Contents

<b>Table of Contents .....</b>	<b>2</b>
<b>Revision History.....</b>	<b>3</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose .....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions .....	1
1.4 Product Scope .....	1
1.5 References.....	1
<b>2. Overall Description.....</b>	<b>1</b>
2.1 Product Perspective.....	2
2.2 Product Functions .....	2
2.3 User Classes and Characteristics .....	2
2.4 Operating Environment.....	<b>Error! Bookmark not defined.</b>
2.5 Design and Implementation Constraints.....	2
2.6 User Documentation .....	2
2.7 Assumptions and Dependencies.....	3
<b>3. External Interface Requirements .....</b>	<b>3</b>
3.1 User Interfaces .....	3
3.2 Hardware Interfaces .....	3
3.3 Software Interfaces .....	3
3.4 Communications Interfaces .....	4
<b>4. System Features .....</b>	<b>4</b>
4.1 Face Recognition .....	4
4.2 Attendance Marking.....	5
4.3 Reports and Exports .....	6
<b>5. Other Nonfunctional Requirements.....</b>	<b>6</b>
5.1 Performance Requirements.....	6
5.2 Safety Requirements.....	6
5.3 Security Requirements.....	7
5.4 Software Quality Attributes .....	7
5.5 Business Rules .....	7
<b>6. Other Requirements .....</b>	<b>7</b>
<b>Appendix A: Glossary .....</b>	<b>7</b>
<b>Appendix B: Analysis Models.....</b>	<b>7</b>
<b>Appendix C: To Be Determined List.....</b>	<b>7</b>

Revision History

Name	Date	Reason For Changes	Version
Soumyadeep Dey	30-07-25	Initial Version	1.0

# 1. Introduction

## 1.1 Purpose

This document describes the software requirements for the Smart Attendance System Using Face Recognition, which automates student attendance using facial recognition technology. This SRS outlines the system's features, constraints, interfaces, and interactions with users.

## 1.2 Document Conventions

- Requirements are labeled as REQ-x.
- High-priority items are in bold.
- Italics denote optional or configurable items.

## 1.3 Intended Audience and Reading Suggestions

- **Developers:** To understand system functionality and constraints.
- **Project Managers:** For scheduling and feature planning.
- **Testers:** To validate system behavior against requirements.
- **Users (Teachers/Admin):** For understanding expected system outputs and use.

## 1.4 Product Scope

The system aims to reduce manual work in taking attendance and increase accuracy by automating it through a real-time camera-based face detection system. Teachers and admins will get real-time attendance records, while students will be verified via facial input.

## 1.5 References

- **OpenCV Documentation:** <https://docs.opencv.org>
- **Python Face Recognition Library:** [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)
- **IEEE 830-1998 SRS Guidelines**

## 2. Overall Description

### 2.1 Product Perspective

This is a **new system** intended to replace traditional/manual attendance systems with biometric face-based attendance tracking. The system integrates camera modules, a web interface, and a backend for record storage.

### 2.2 Product Functions

- Register teacher and student accounts
- Upload and manage facial data
- Detect and recognize faces from webcam
- Mark attendance if face is verified
- Generate attendance reports per day/month/subject
- Admin dashboard for data analysis

### 2.3 User Classes and Characteristics

- **Admin:** Full access to system, user management, and report generation
- **Teacher:** Can view/add student face data and mark attendance
- **Student:** Limited access to their own attendance records
- **Admin:** Full access to system, user management, and report generation
- **Teacher:** Can view/add student face data and mark attendance
- **Student:** Limited access to their own attendance records

### 2.5 Design and Implementation Constraints

- Face must be captured in good lighting
- Minimum webcam resolution: 640x480
- Passwords must be stored using hashing (bcrypt)
- Data must be encrypted when stored or transmitted

## **2.6 User Documentation**

- Online help documentation.
- Admin/User training manual (PDF).
- System setup guide.

## **2.7 Assumptions and Dependencies**

- Users have access to webcams.
- Lighting conditions are good enough for face detection.
- Accurate initial training data is available.
- Internet connectivity for cloud sync (if used).

# **3. External Interface Requirements**

## **3.1 User Interfaces**

- Login Page for admins/teachers.
- Dashboard with charts for attendance reports.
- Face Registration via webcam or file upload.
- Live Face Recognition window for attendance.
- Consistent navigation menu and error feedback dialogs.

## **3.2 Hardware Interfaces**

- Interface with webcam for capturing live image data
- USB camera compatibility ensured.

## **3.3 Software Interfaces**

- MySQL/MongoDB.
- Python Face Recognition library.
- OpenCV.
- Flask/Django APIs.

### 3.4 Communications Interfaces

- RESTful APIs for data communication between frontend and backend.
- HTTPS for secure communication.
- Optional email integration for absence alerts

## 4. System Features

This section outlines the functional capabilities of the Smart Attendance System Using Face Recognition. These features are designed to meet the primary objective of automating the attendance process through real-time facial recognition while maintaining security, accuracy, and ease of use.

The system provides features for different user roles including administrators, teachers, and students. Core functionalities include student face registration, face detection and recognition, attendance marking, report generation, and system management through an intuitive user interface. Each feature is described in terms of its purpose, usage priority, and functional behavior.

The following features have been grouped logically based on the major services provided by the system. Each feature is detailed in subsequent subsections, with an emphasis on the user interactions, stimulus-response behavior, and corresponding functional requirements.

### 4.1 Face Recognition

#### 4.1.1 Description and Priority

- Students' faces are registered via webcam or image upload
- **Priority: High**

#### 4.1.2 Stimulus/Response Sequences

- User selects student → Captures photo → System stores encoded facial features

#### 4.1.3 Functional Requirements

- **REQ-1:** System must allow image upload or capture
- **REQ-2:** System must encode and store face
- **REQ-3:** System must prevent duplicate registration

## 4.2 Attendance Marking

### 4.2.1 Description and Priority

- System automatically marks present students via live webcam
- **Priority: High**

### 4.2.2 Stimulus/Response

- User starts session → Webcam scans → Faces matched → Attendance updated

### 4.2.3 Functional Requirements

- **REQ-4:** Match live face with stored data
- **REQ-5:** Update record in attendance table
- **REQ-6:** Prevent duplicate entries for the same session

## 4.3 Reports and Export

### 4.3.1 Description and Priority

- Teachers/Admins can view and export attendance
- **Priority: Medium**

### 4.3.2 Functional Requirements

- **REQ-7:** System must generate reports by date, student, subject
- **REQ-8:** Export as PDF or CSV
- **REQ-9:** Send monthly report via email (*Optional*)



## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

- Face detection must occur in under 2 seconds
- System should handle 50+ concurrent users
- Daily attendance report generation  $\leq 5$  sec

### **5.2 Safety Requirements**

- System logs must record access and attendance attempts
- Ensure physical safety by not requiring biometric hardware

### **5.3 Security Requirements**

- Login must use strong password hashing
- Session timeout after 10 minutes
- Role-based access enforced
- HTTPS enforced for all connections

### **5.4 Software Quality Attributes**

- Usability: Minimal training needed
- Reliability: Must work in >95% of lighting conditions
- Maintainability: Modular code structure
- Scalability: Should support institution-wide deployment

### **5.5 Business Rules**

- Only registered students can be marked present
- Only teachers can initiate attendance sessions
- One attendance per class per student per day

## **6. Other Requirements**

- Weekly auto-backup of attendance database

- Multi-language support (Optional)
- Integration with ERP or LMS systems (Optional)
- Facial data should comply with local data protection laws

## Appendix A: Glossary

- Face Encoding: A numerical representation of facial features
- OpenCV: Computer vision library
- Flask: A lightweight Python web framework
- LDAP: Lightweight Directory Access Protocol (*Optional future integration*)

## Appendix B: Analysis Models

- Use Case Diagram: Admin, Teacher, Student actors
- ER Diagram: Tables for users, attendance, face, data
- Data Flow Diagram: Face input → Process → Match → Log result

## Appendix C: To Be Determined List

- TBD-1: Final camera hardware configuration for deployment
- TBD-2: Whether mobile app integration is planned
- TBD-3: Choice of SMS/email provider for alerts