

# **Software Requirements Specification**

**for**

## **<Attendance Management System>**

**Version 1.0 approved**

**Prepared by <Sunny Kaushik, Samarjeet Wankhade, Saanvi Vishal, Varshith Vattikuti>**

**<IIIT Bangalore>**

**<27/10/2023>**

# 1. Introduction

## 1.1 Purpose

*This SRS (Software Requirements Specification) document aims to outline the software requirements for the Attendance Management System through Facial Recognition system. It serves as a comprehensive guide to the system's features and functionalities.*

## 1.2 Intended Audience and Reading Suggestions

*This document is designed for a diverse audience, including developers, project managers, marketing personnel, users, testers, and documentation writers. It is structured with a preference for readers to navigate it primarily through the table of contents. **The target audience will be an organization which needs to keep track of their employees' attendance.** The rest of the document covers the compatible environment, user documentations and functional and non-functional requirements.*

## 1.3 Product Scope

*The Attendance Management System is designed to enable personnel responsible for tracking attendance within an organization to maintain attendance records and essential student information from a computer. Additionally, the system offers various functionalities, including editing attendance, viewing statistics, and making notes related to attendance-related elements. **The aim is to provide a convenient and portable solution for personnel, such as professors or administrators, to streamline attendance record management and access attendance statistics within their organization.***

# 2. Overall Description

## 2.1 Product Perspective

*This project forms an integral component of our coursework in CS-303P Software Engineering at IIIT Bangalore. Its objective is to streamline the attendance tracking process for students by implementing a Facial Recognition System. This innovative solution promises to not only eliminate the hassles associated with traditional methods but also make the entire process significantly easier.*

## 2.2 Product Functions

- *Enrollment and Database Management:*
  - *Capture and store facial images of users.*
  - *Maintain a database of enrolled users.*
  - *Update, add, or delete user records as needed.*

- *Facial Recognition:*

Page 2

- *Perform real-time facial recognition for attendance tracking.*
  - *Match captured images to the enrolled student's facial data.*
  - *Handle variations in lighting, angles, and facial expressions.*
- *Attendance Tracking:*
  - *Automatically record attendance based on recognized faces.*
  - *Generate reports and logs of attendance data.*
- *User Interface:*
  - *Provide an intuitive, user-friendly interface for administrators.*
  - *Offer a simple and efficient method for users to check-in.*
- *Notifications and Alerts:*
  - *Send notifications of attendance anomalies or discrepancies.*
  - *Alert administrators of system errors or failures.*

## 2.3 User Classes and Characteristics

Here are the User Classes and their Characteristics that are expected to be implemented.

- *Administrators:*
  - *Characteristics: Highly privileged users responsible for system management.*
  - *Educators responsible for attending educational institutions.*
- *Functions:*
  - *Enroll and manage users.*
  - *Take attendance for their classes using the facial recognition system.*
- *Students or Employees:*
  - *Characteristics: End-users attending classes or working regularly.*
- *Functions:*
  - *Use the system to check in and record their attendance.*
  - *View their own attendance recorded or not.*
  - *Receive a prompt regarding their attendance.*

## 2.4 Operating Environment

Here are Some of the Environment necessary to make sure the system will operate ideally.

*Hardware Platform:*

*The software is designed to operate on the following hardware platform: Server Hardware:*

- *The software will run on personal computers or web-based Systems and send data to a cloud-based storage namely Firebase by Google.*
- *The computer should have a multicore processor for handling concurrent requests efficiently as using an AI model is quite hefty and needs lots of resources to process.*
- *Sufficient RAM to ensure smooth operation System.*
- *A high-resolution Web Camera to capture the Image.*
- *A screen display (for server User Interactions).*

- Adequate storage space for data storage and system logs on Firebase cloud Server.

#### *Operating System and Versions:*

- Any General Operating System should work with it.

#### *Software Components and Applications:*

*Page 3*

*The software must coexist peacefully with various other software components and applications, which may include:*

- Python, Python Interpreter and Python Runtime Environments and Libraries like PyCharm, cmake, dlib, face-recognition, cvzone, numpy, opencv-python version 4.5.4.60 • Python Interpreter
- Internet Connectivity Interface
- Database Systems: Firebase
- Fire Base APIs

#### *Development Tools:*

- Compatibility with development tools and IDEs (Integrated Development Environments) that developers use to work on software such as Visual Studio Desktop Development Version. Version

#### *Control Systems:*

- Compatibility with version control systems (e.g., Git) used by development teams for source code management.

## 2.5 Design and Implementation Constraints

Constraints for this project can be stated below:

#### *Regulatory Policies and rights to use the needed tools:*

- Data handling rights and privacy rights along with rights to use the tools to create this platform. Use of specific programming languages, frameworks, and development tools. •

*Hardware Limitations: Specific hardware requirements, such as limited memory or processing power, can constrain the software design and functionality.*

- Interfaces to Other Applications: Integration with existing systems or third-party applications may require adherence to predefined data formats, communication protocols, or APIs specifically firebase.

- Database Requirements: **Requirements for data storage and management can limit the choice of database systems and their configurations.**

- Parallel Operations: If the system needs to support concurrent or parallel operations, developers may have to follow specific design patterns and technologies to ensure synchronization and consistency.

- Communications Protocols: Protocols for communication with other systems or devices, such as APIs, networking protocols, or messaging formats, may be predefined. •

*Environmental Considerations: Environmental factors, such as the need for the software to operate in specific conditions (e.g., proper lighting), can impose constraints on hardware and software choices.*

- A resolution Camera is also required with capabilities to integrate with system.

## 2.6 Assumptions and Dependencies

Here are the Assumptions and Dependencies necessary for this project.

- *Third-Party Facial Recognition API:*
  - *Assumption: The project relies on a third-party facial recognition API for its core functionality.*
  - *Dependency: The availability and continued support of this API are crucial to the project's success. Any changes or discontinuation of the API could impact the system's functionality.*
- *Hardware Requirements:*
  - *Assumption: The SRS assumes a standard set of hardware components for the deployment environment (e.g., cameras, server infrastructure).*

*Page 4*

- *Dependency: Any deviations or changes in the hardware infrastructure may require adjustments in the software to ensure compatibility and performance.*
- *Network Connectivity:*
  - *Assumption: The SRS assumes a stable network connection for real-time facial recognition.*
  - *Dependency: Network outages, latency, or inadequate bandwidth could affect system performance and user experience.*
- *Environmental Factors:*
  - *Assumption: The project may assume a controlled environment for facial recognition (e.g., indoor, well-lit conditions).*
  - *Dependency: Environmental factors, such as outdoor usage or poor lighting conditions, could affect the system's accuracy and reliability.*

### 3. External Interface Requirements

#### 3.1 User Interfaces

*The attendance management system should provide interface which will aid in taking the attendance through the facial recognition system. The interface will consist of a real time face detection system, checking for the target subject whether the attendance is marked or not. If the user is not registered into the system, a pop will appear notifying them of the same, which must be registered by the administrator into the database. The interface will also display some of the characteristics (like the stream, roll number etc.) of the subject to verify for the right student marking the attendance.*

#### 3.2 Hardware Interfaces

*Server Side:*

- *Operating system: Windows 8 or above, MAC OS, UNIX, LINUX*
- *Processor: 3 GHz or higher*
- *RAM: 4GB or higher*
- *Hard Drive: 1 TB or higher*

*These are not strictly bounded specs*

*Client Side:*

- *Operating system: Windows 8 or above, MAC OS, UNIX, LINUX*

- *Processor: 1 GHz or higher*
- *RAM: 2GB or higher*

### 3.3 Software Interfaces

- *The software will transmit the attendance for the organization to a real-time database.*
- *If the user is not registered with the organization or the database does not contain the record of that employee/student, then it needs to be registered by the individual.*
- *The user will not be allowed to modify the attendance records at any time.*
- *The software will communicate with the real time database to allow a user to enter their attendance through the facial recognition system.*

Page 5

## 4. System Features

### 1. User Registration and Management:

- *Feature 1.1: User Registration*
- The system should allow administrators to add and register users.
- Users can provide basic information like name, ID, and photograph.
- Facial templates should be generated and stored for each user.
- *Feature 1.2: User Access Control*
- Different user roles (e.g., administrators, teachers, students) should have distinct access levels.
- Administrators can manage user accounts, while students may have limited access.

### 2. Attendance Tracking:

- *Feature 2.1: Real-time Attendance Tracking*
- The system should capture and recognize faces in real-time during class or work hours.
- The attendance status (e.g., present, absent) should be automatically recorded.

#### *Feature 2.2: Attendance History*

- The system should maintain a history of attendance records for each user.

Users and administrators can access and review past attendance data.

### 3. Facial Recognition:

- *Feature 3.1: Face Enrollment*
- The system should allow users to enroll their faces for recognition.
- *Feature 3.2: Face Recognition*
- The system should accurately recognize registered faces during attendance checks.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

#### 1. Recognition Speed:

- a. *Requirement:* The system must recognize and record attendance within

seconds of capturing a face.

- b. *Rationale*: Quick recognition is essential to minimize disruptions and maintain an efficient attendance process.

2. Accuracy:

- a. *Requirement*: The system must achieve an accuracy rate of at least 99% in facial recognition.
- b. *Rationale*: High accuracy is essential to prevent false positives or negatives, which could lead to attendance inaccuracies.

3. Response Time:

- a. *Requirement*: The system should respond to user requests (e.g., queries and reports) within 3 seconds.
- b. *Rationale*: Fast response times are necessary for a seamless user experience.

Page 6

## 5.2 Safety and Security Requirements

- **Data Protection**: The system must safeguard sensitive biometric data (facial images) from loss, theft, or unauthorized access. Data should be encrypted both in transit and at rest.
- **Minimize False Positives/Negatives**: The system should minimize false recognition errors to prevent unauthorized individuals from marking attendance or legitimate users from being denied access.
- **Access Control**: Implement strict access controls to ensure only authorized personnel can access the system settings and data.
- **User Identity Authentication**: Define robust methods for verifying the identity of individuals during the enrollment process to prevent impersonation.

## 5.3 Software Quality Attributes

- **Correctness**: The system should achieve a facial recognition accuracy rate of at least 95% in various lighting and environmental conditions.
- **Robustness**: It should maintain functionality even when faced with low-quality images, such as those with partial face occlusions, glasses, or headgear.
- **Scalability**: The system should handle a growing number of users and records without significant performance degradation.
- **Security**: The system should ensure the privacy and security of user data and comply with relevant data protection regulations. It should have robust encryption and authentication mechanisms.
- **Reusability**: The system should be designed with modular components to promote code reusability for future projects.

## 5.4 Business Rules

- **User Enrollment and Registration**:
  - **Principle**: Only authorized administrators or designated personnel should be

able to enroll individuals into the system.

- Functionality Implication: The system should have user roles, with administrators having the privilege to add new individuals and assign them to specific groups or roles.

- Attendance Recording:

- Principle: The system should record attendance when an individual's face is recognized.

- Functionality Implication: The system must have facial recognition capabilities and the ability to match recognized faces to registered users.

- Access to Attendance Data:

- Principle: Access to attendance records should be role-based, with different levels of access for administrators, teachers, and students.