Software Requirements Specification (SRS)

# Project: Virtual Attendance System

## 1. Introduction

### Purpose

The purpose of this document is to define the requirements and specifications for a Virtual Attendance System that ensures accurate and secure student attendance tracking in academic environments. This system leverages dynamic scanner codes and Wi-Fi validation to confirm physical presence in class. It helps prevent proxy attendance and simplifies attendance management for faculty.

### Scope

The system allows teachers to generate dynamic scanner codes every 5 seconds, which students must scan through their mobile applications. The scan will be validated only if the student's device is connected to the institution's official Wi-Fi. Admins can view attendance reports, manage users, and export data. The application supports web and mobile platforms and is intended for deployment within college campuses.

### Overview

This document outlines the complete Software Requirements Specification (SRS) for the Virtual Attendance System, covering functional and non-functional requirements, system features, user interfaces, performance benchmarks, and design considerations for building a robust and scalable attendance tracking solution.

## 2. General Description

### Functions

The main functions of the system include secure login for all users, dynamic QR/token code generation, attendance validation using scanned input and SSID detection, attendance logging with timestamps, and report generation based on filters such as date, course, or student.

### User Community

The system will serve three main user groups: Students (who scan codes to mark attendance), Teachers (who initiate sessions and display dynamic codes), and Admins (who manage users, sessions, and reporting). Each role has specific access rights defined by the system.

## 3. Functional Requirements

### Possible Outcomes

Possible results of a scan include: successful attendance mark, invalid code, scan expired, wrong network detected, or duplicate scan attempt. Each outcome triggers a specific system response.

### Ranked Order

1. Accurate scanner code matching.  
2. Network SSID validation.  
3. Secure login/authentication.  
4. Real-time dashboard updates.  
5. Reliable logging and report export.

### Input-Output Relationship

Inputs include scanner code and current Wi-Fi SSID. The system processes these inputs, validates them against active session and expected network, and outputs a response indicating whether attendance was successfully marked or rejected.

## 4. User Interface Requirements

### Software Interfaces

The system will consist of a responsive web portal for teachers and admins, and an Android mobile application for students. The backend will provide APIs for user login, scan validation, and reporting.

### Examples

Teachers will see a dashboard with a start session button and live code display. Students will have a scan screen showing countdown timers and network connection status. Admins can view analytics and logs.

## 5. Performance Requirements

### Response Time

Scan validation should be processed within 1–2 seconds to ensure a smooth classroom experience. System response should be near real-time, even under load.

### Throughput

The system should be capable of handling at least 100 concurrent scans per class session with no significant delays. Backend optimization and lightweight APIs are essential.

### Scalability

The application must support multi-classroom and multi-department use. It should be scalable to a campus-wide or university-wide implementation with minimal reconfiguration.

## 6. Non-Functional Attributes

### Usability

The UI will be user-friendly, requiring minimal training for students and staff. Clear icons, tooltips, and step-by-step guidance will be integrated.

### Reliability

System uptime should exceed 99% during class hours. In case of failure, attendance data should be cached locally and synced when connection is restored.

### Security

All user sessions will be encrypted. Attendance will be validated using time-sensitive codes and verified Wi-Fi SSIDs. Attempts from non-college networks will be blocked.

## 7. Schedule and Budget

### Timeline

Development will take approximately 6–8 weeks. Phases include: Requirements (Week 1), Design (Week 2–3), Implementation (Week 4–6), Testing (Week 7), and Final Review/Deployment (Week 8).

### Cost Estimate

As a student project, development cost is minimal. Open-source libraries and tools will be used. Estimated cost: ₹0–500 for optional deployment or hosting.

## 8. Appendices

### Supplementary Information

Dynamic codes will be generated using a secure random number/token generator. Valid SSIDs will be predefined in system settings. System logs will include device ID, IP address, and timestamp.

### Glossary

- SSID: Wi-Fi network name  
- QR Code: A machine-readable 2D code  
- Token: A temporary numeric code  
- Session: A teacher-initiated attendance time window  
- Scan: Student action to capture displayed code