Product Requirements Document (PRD)

Smart Multi-Layered Attendance Verification System (SMAVS)

1. Overview

The Smart Multi-Layered Attendance Verification System (SMAVS) is a secure mobile application designed to eliminate impersonation and proxy attendance in university classrooms. It leverages multi-factor verification — including code entry, biometric checks, device validation, and classroom presence confirmation — to ensure only genuine students are marked present.

2. Goals and Objectives

- Primary Goal: Prevent impersonation and proxy attendance in classrooms.
- Objectives:
• Provide a secure and seamless attendance-taking process.
• Integrate multiple verification layers: code, identity, device, and location.
• Ensure accountability for both students and lecturers.
• Maintain lecturer control with flexibility to repeat verification during class.

3. Key Features

• Role-Based Dashboards (Single App):
- Students: Enter code, selfie verification, IMEI check, receive confirmation.
- Lecturers: Generate codes, monitor results, handle flagged cases, export reports.
- Admins: Manage accounts, oversee logs, approve IMEI resets.
• Attendance Workflow
• Admin Web Dashboard

4. User Roles

• Students: Verify attendance with code + selfie + IMEI, receive confirmation.
• Lecturers: Generate codes, view verifications, handle flags, export reports.
• Administrators: Manage accounts, approve IMEI resets, monitor logs and system.

5. Functional Requirements

FR1. Students must log in with verified credentials before using app.
FR2. Lecturer must be able to generate a unique code per session.
FR3. Students must input code to trigger attendance verification.
FR4. System must capture and verify selfie image.
FR5. System must capture and verify classroom presence via back camera.
FR6. IMEI must be logged and matched with database.
FR7. System must provide immediate attendance confirmation or flag discrepancies.
FR8. Lecturer can repeat verification at any point.
FR9. Attendance reports must be exportable.

6. Non-Functional Requirements

• Security: Data encryption, device binding
• Scalability: Support 10,000+ concurrent users
• Performance: Verification in <5 seconds
• Reliability: 99.9% uptime
• Usability: Intuitive, offline support

7. Technical Stack

• Frontend: React Native / Flutter
• Backend: Node.js or Python (FastAPI / Django)
• Database: PostgreSQL, Redis
• Biometric Verification: OpenCV, TensorFlow Lite or AWS Rekognition
• Hosting: AWS / GCP / Azure
• Security: JWT, HTTPS, AES-256 encryption

8. Constraints & Assumptions

• Students and lecturers need smartphones with internet
• IMEI check requires Android compatibility
• University database access required
• Classroom presence assumed via back-camera validation

9. Success Metrics

• >95% accuracy in preventing proxy attendance
• <5 seconds verification time
• >80% adoption rate
• Positive feedback (>4/5 rating)

10. Risks & Mitigation

• Fake photos → Liveness detection
• Device theft → Admin IMEI reset process
• Poor connectivity → Offline code entry with sync

11. Timeline (Proposed)

Requirements: 2 weeks
UI/UX: 3 weeks
Backend: 6 weeks
Mobile App: 8 weeks
Biometric Integration: 4 weeks
Testing: 3 weeks
Deployment: 2 weeks
Total: ~7 months

12. Deliverables

• Unified Mobile App (Android & iOS) with role-based dashboards
• Admin Web Dashboard
• Backend APIs and Database
• Attendance Reports
• Documentation & Training Materials