ClassPulse PDR - Smart Classroom Attendance System

Table of Contents

Executive Summary

ClassPulse is an intelligent attendance tracking system that combines Raspberry Pi device detection, mobile app integration, camera-based headcount verification, and time-based attendance logic to create a comprehensive anti-proxy classroom monitoring solution.

Key Innovation: Duration-based attendance tracking with headcount verification prevents proxy attendance while maintaining student privacy.

System Architecture

1. Mobile Application (Flutter)

User Flow:

- **Step 1**: App Launch & Splash Screen
- **Step 2**: One-Time Registration Form
  - Student personal details (name, email, phone, photo)
  - Academic information (course, year, section)
  - Automatic device identifier collection (MAC, UUID generation)
  - Comprehensive permission requests (Bluetooth, Wi-Fi, Location, Camera)
- **Step 3**: Home Screen with Timetable Management
  - Personal class schedule display
  - PDF/Image upload for timetable updates (OCR integration)
  - **Dynamic Configuration Panel** (Demo Mode)
    - Raspberry Pi network settings (IP, port)
    - Detection thresholds and scan intervals
    - Geofencing parameters
    - Camera module configuration
    - Attendance duration rules

2. Backend Architecture

- **Firebase Integration**: Real-time student data synchronization
- **Local Storage**: SQLite for offline functionality
- **Authentication**: OAuth/Phone-based secure login
- **Data Flow**: Bidirectional sync between app, Firebase, and Raspberry Pi

3. Raspberry Pi Core System (3B+)

Hardware Components:

- Raspberry Pi 3B+ (Central Controller)
- Camera Module v2 (Headcount verification)
- Wi-Fi Adapter (Monitor mode for probe requests)
- GSM Module SIM800L (SMS notifications)
- MicroSD 16GB+ (OS and database storage)
- Power Supply 5V/3A

Software Stack:

- Raspberry Pi OS Bookworm

- Python 3 with libraries: bleak, scapy, opencv-python, flask
- SQLite database
- Nginx web server

4. Web Dashboard

Features:

- Student management and attendance records
- Real-time headcount vs device count comparison
- Proxy alert system
- Timetable management
- Statistical reports and data export
- Manual attendance override capabilities

Core Functionality

Time-Based Attendance Logic

Duration Thresholds:

- **Present**: ≥45 minutes detected during class
- **Proxy**: 15-44 minutes detected (flagged for review)
- **Absent**: <15 minutes detected

Anti-Proxy Measures

1. **Duration Tracking**: Continuous device monitoring throughout class
2. **Headcount Verification**: Camera-based people detection at class start
3. **Device Count Matching**: Real-time comparison of detected devices vs visual headcount
4. **Geofencing**: RSSI-based proximity verification (-70dBm threshold)
5. **UUID Tracking**: Persistent identifiers to counter MAC randomization

Camera Integration

- **Trigger**: Automated capture at class commencement
- **Processing**: OpenCV-based people detection and counting
- **Verification**: Cross-reference headcount with device detection count
- **Alert System**: Flag discrepancies for manual review

Technical Specifications

Database Schema

```
-- Students table with extended fields
CREATE TABLE students (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    email TEXT,
    phone TEXT,
    mac_address TEXT UNIQUE NOT NULL,
    device_id TEXT,
    uuid TEXT UNIQUE,
    fcm_token TEXT,
    course TEXT,
    year TEXT,
    section TEXT,
    photo_url TEXT
);

-- Attendance with duration tracking
CREATE TABLE attendance (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    student_id INTEGER,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    duration INTEGER,
    status TEXT, -- 'present', 'proxy', 'absent'
    rssi INTEGER,
    detection_method TEXT, -- 'bluetooth', 'wifi', 'both'
    FOREIGN KEY(student_id) REFERENCES students(id)
);

-- Headcount verification
CREATE TABLE headcount (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    count INTEGER,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    image_path TEXT,
    verified BOOLEAN DEFAULT 0,
    discrepancy_flag BOOLEAN DEFAULT 0
);

-- Dynamic configuration
CREATE TABLE app_config (
    id INTEGER PRIMARY KEY,
    pi_ip TEXT,
    pi_port INTEGER,
    scan_interval INTEGER,
    rssi_threshold INTEGER,
    duration_present INTEGER DEFAULT 45,
    duration_proxy_min INTEGER DEFAULT 15,
    geofence_radius REAL,
    camera_enabled BOOLEAN DEFAULT 1
);
```

Flutter Dependencies

```
dependencies:
  flutter:
    sdk: flutter
  device_info_plus: ^9.0.0
  flutter_bluetooth_serial: ^0.4.0
  shared_preferences: ^2.2.0
  uuid: ^4.0.0
  firebase_messaging: ^14.6.0
  firebase_auth: ^4.7.0
  cloud_firestore: ^4.8.0
  image_picker: ^1.0.0
  file_picker: ^5.3.0
  permission_handler: ^10.4.0
  geolocator: ^9.0.0
  http: ^1.1.0
```

Raspberry Pi Setup Commands

```
# System preparation<a></a>
sudo apt update
sudo apt install python3-pip python3-dev libpcap-dev nginx
sudo pip3 install bleak pyserial schedule scapy flask flask-cors opencv-python firebase-a

# Wi-Fi monitor mode setup<a></a>
sudo iwconfig wlan1 mode monitor
sudo ifconfig wlan1 up

# Camera module enable<a></a>
sudo raspi-config
# Interface Options &gt; Camera &gt; Enable<a></a>
```

Configuration Management

Dynamic App Configuration Panel

Network Settings:

- Raspberry Pi IP address
- Communication port
- Connection timeout values
- Retry attempts

Detection Parameters:

- Bluetooth scan interval (default: 60 seconds)
- Wi-Fi probe request timeout
- RSSI threshold for geofencing

- Device detection sensitivity

Attendance Rules:

- Present duration threshold (45 minutes)
- Proxy detection range (15-44 minutes)
- Absent threshold (<15 minutes)
- Grace period for late arrivals

Camera Configuration:

- Headcount capture frequency
- Image processing sensitivity
- Verification tolerance percentage
- Storage and retention settings

Security and Privacy

Data Protection

- MAC address encryption at rest
- UUID-based anonymous tracking
- Firebase security rules implementation
- Local data encryption on device

Permissions and Consent

- Explicit consent forms during registration
- Clear data usage explanations
- Opt-out mechanisms for privacy-conscious students
- GDPR compliance measures

Access Control

- Role-based dashboard access (teacher, admin)
- API endpoint authentication
- Secure communication protocols
- Regular security audits

Implementation Timeline

Phase 1: Core Development (2 weeks)

- Flutter app with registration and configuration
- Basic Raspberry Pi detection system
- SQLite database implementation
- Firebase integration

Phase 2: Advanced Features (2 weeks)

- Camera module integration
- Headcount verification system
- Web dashboard development
- Anti-proxy logic implementation

Phase 3: Testing and Optimization (1 week)

- Classroom environment testing
- Performance optimization
- UI/UX refinements
- Documentation completion

Cost Analysis

Hardware Costs (Per Classroom)

- Raspberry Pi 3B+: ₹4,500
- Camera Module v2: ₹2,500
- Wi-Fi Adapter: ₹1,500
- GSM Module: ₹1,200
- MicroSD Card 32GB: ₹800
- Power Supply & Accessories: ₹1,500
- **Total Hardware**: ₹12,000

Software Costs

- Firebase Usage: ₹500/month (estimated for 100 students)
- Development Tools: One-time setup
- Maintenance: Minimal ongoing costs

Risk Assessment

Technical Risks

- **MAC Address Randomization**: Mitigated by UUID system
- **Wi-Fi Interference**: Addressed through adaptive scanning
- **Power Management**: External GSM module power supply
- **Camera Privacy Concerns**: Anonymous processing, no face recognition

Operational Risks

- **Student Compliance**: Push notifications for connectivity reminders
- **Hardware Failure**: Redundant detection methods (Bluetooth + Wi-Fi)
- **Network Connectivity**: Local operation with periodic sync

Success Metrics

Primary KPIs

- Attendance accuracy: >95%
- Proxy detection rate: >85%
- System uptime: >99%
- Student adoption: >90%

Secondary Metrics

- Configuration change frequency
- Dashboard usage statistics
- Camera verification accuracy
- Teacher satisfaction scores

Future Enhancements

Planned Features

- Multi-classroom deployment
- Integration with LMS systems
- Advanced analytics and reporting
- Machine learning for behavioral patterns
- Cloud-based scalability

Research Opportunities

- Thermal imaging integration
- Voice recognition attendance
- IoT sensor fusion
- Blockchain verification system

Conclusion

ClassPulse represents a comprehensive solution to automated attendance tracking that balances technological innovation with practical classroom needs. The dynamic configuration system ensures flexibility for testing and deployment, while robust anti-proxy measures maintain system integrity. The combination of device detection, camera verification, and time-based logic creates a reliable attendance system suitable for modern educational environments.

The system's modular architecture allows for easy expansion and integration with existing educational technology infrastructure, making it a valuable addition to smart classroom initiatives.