# Mobile-Based Attendance Management System | Group 24 | Task 2 report

- Research suggests that a mobile-based attendance system with facial recognition and geofencing can improve accuracy and efficiency in schools.
- It seems likely that combining GPS for location checks and on-device facial recognition can achieve quick check-ins, possibly within 5 seconds.
- Evidence leans toward Flutter for building the app, as it works well on both Android and iOS, with tools like Google ML Kit for facial recognition and geolocation plugins for geofencing.

## Why This System Matters

Taking attendance in schools can be slow and prone to mistakes, like students signing in for friends. A mobile app using facial recognition to identify students and geofencing to confirm they're in class could make things faster and more reliable. This project aims to create such an app, designed to be easy for students and teachers to use while keeping records secure.

## How We Gathered Information

We talked to students, teachers, and school staff to understand what they need. A survey asked about ease of use, speed, and comfort with facial recognition. We also held interviews, brainstormed ideas, and looked at other attendance systems to learn what works and what doesn't. This helped us plan an app that meets everyone's needs.

## Key Findings

Students generally find the idea of a mobile app easy to use, with many comfortable using facial recognition, though some worry about network issues slowing things down. Teachers want real-time attendance data and simple tools to manage classes. The app will use Flutter, which is great for building user-friendly apps, and include features like quick check-ins and location verification to address these concerns.

# Comprehensive Requirement Gathering Report for Mobile-Based Attendance Management System

## Introduction

This report details the requirement gathering process for a mobile-based attendance management system designed for higher education institutions. The system leverages facial recognition and geofencing technologies to provide an efficient, secure, and automated solution for tracking student attendance. Developed using Flutter for cross-platform compatibility, the app aims to address inefficiencies in traditional methods like manual sign-ins and RFID cards, which are prone to errors and manipulation. By integrating insights from stakeholders and research, this report ensures the system meets user needs while maintaining high performance and reliability.

## Background and Problem Statement

Traditional attendance methods, such as paper sign-in sheets and RFID cards, are time-consuming, susceptible to proxy attendance, and often result in delayed record access. Research highlights that these systems struggle with accuracy and efficiency, particularly in large academic settings (Face Recognition based Attendance Management System). Mobile devices, now widely accessible, offer a solution by enabling advanced features like facial recognition for secure identification and geofencing for location verification. This project seeks to develop a mobile application that combines these technologies to deliver real-time, accurate attendance tracking with minimal human intervention.

## Stakeholder Identification

The system's success depends on addressing the needs of key stakeholders, identified as follows:

- **Students**: Primary users who will check in using the app, requiring an intuitive interface and quick process.
- **Instructors**: Monitor attendance and manage sessions, needing real-time data and filtering options.
- **Administrative Staff**: Use attendance records for reporting and analytics, requiring reliable and accessible data.
- **IT Department**: Maintain the system, ensuring scalability, security, and integration with existing infrastructure.
- **University Management**: Approve and fund the project, expecting alignment with institutional goals and cost-effectiveness.

Engaging these groups ensures the system is practical and widely adopted.

## Requirement Gathering Techniques

To capture comprehensive requirements, multiple techniques were employed:

- **Surveys**: A survey collected quantitative and qualitative feedback from students and other users on ease of use, performance acceptability, and comfort with biometrics. Responses provided critical insights into user expectations and concerns.
- **Interviews**: One-on-one discussions with instructors and IT staff explored specific needs, such as session management tools and technical reliability.
- **Brainstorming**: Collaborative sessions with stakeholders and developers generated innovative ideas, including geofencing for location verification and on-device processing for privacy.
- **Reverse Engineering**: Analysis of existing systems, such as Timeero and Truein (7 Best Face Recognition Attendance System in 2024 - Timeero), identified strengths (e.g., automation) and weaknesses (e.g., cloud-based latency) to inform design improvements.

This multi-faceted approach ensured a thorough understanding of requirements from diverse perspectives.

# Data Gathering

Data was collected from various sources, with the primary input being a user survey. The survey included questions on usability, performance, biometric comfort, and suggestions for improvements. Below is a sample of the collected data:

| Timestamp | Role | Ease of Use | Performance Acceptability | Comfort with Biometrics | Suggested Features | Challenges Foreseen |
|---|---|---|---|---|---|---|
| 2025/04/12 10:07:53 PM | Student | Easy | Neutral | 3 | Request permission option | Network disturbances may slow check-ins |
| 2025/04/13 1:52:54 AM | Student | Very Easy | Acceptable | 5 | Handle location changes per timetable | Tracking correct location if class moves |
| 2025/04/14 8:28:45 AM | Other | Easy | Not Acceptable | 4 | None | None |

## Survey Insights

- **Ease of Use**: Most respondents rated the app as "Easy" or "Very Easy," though some were neutral, suggesting a need for intuitive design.
- **Performance Acceptability**: Responses varied, with concerns about network reliability affecting speed perceptions.
- **Comfort with Biometrics**: Ratings ranged from 3 to 5 (out of 5), indicating moderate to high acceptance, though some hesitation exists.
- **Suggestions and Challenges**: Users requested features like permission requests and flexibility for location changes, while highlighting network issues and location tracking accuracy as potential barriers.

Additional data came from:

- **Interviews**: Instructors emphasized real-time attendance views and manual override options, while IT staff focused on system uptime and data security.
- **Existing Systems**: Analysis revealed slow check-in times and lack of location verification in many solutions, reinforcing the need for geofencing (Implementation Of Mobile Attendance Application Using Geo-Fence Technique).

## Data Cleaning

To ensure data reliability, the survey responses were cleaned as follows:

- **Incomplete Responses**: Entries missing key fields (e.g., role or ratings) were removed.
- **Standardization**: Text responses were normalized (e.g., "Easy" to lowercase) for consistency.
- **Outlier Handling**: Extreme ratings inconsistent with comments were reviewed and adjusted if necessary.
- **Anonymization**: Email addresses were removed to protect privacy, aligning with data protection standards.

This process produced a clean dataset for accurate analysis.

## User Reluctance Assessment

Survey responses highlighted factors influencing user adoption:

- **Biometric Comfort**: An average rating of approximately 4.2 out of 5 suggests general acceptance, though some users expressed mild concerns, possibly due to privacy or unfamiliarity.
- **Ease of Use**: Variability in responses indicates a need for a simple interface, with onboarding support for less tech-savvy users.
- **Performance Concerns**: Comments like "network usually disturbs" suggest technical reliability is critical to prevent frustration.
- **Feature Requests**: Suggestions for permission options and timetable-based location adjustments reflect a desire for control and flexibility.

To address reluctance, the system will include clear privacy policies, user-friendly design, and robust performance optimizations.

## Technology Selection: Flutter

Flutter was chosen for its cross-platform capabilities, high performance, and rich UI components, making it ideal for academic settings (Comparative Study of Flutter with Other Mobile App Development Frameworks). Key integrations include:

- **Facial Recognition**: Google ML Kit (Google ML Kit for Flutter) enables on-device processing, achieving inference times around 160ms, similar to MobileFaceNet in TensorFlow Lite (Recognizing Face in Android using Deep Neural Network + TensorFlow Lite).
- **Geofencing**: The `geolocator` plugin (Geolocator for Flutter) leverages GPS for location verification, ensuring students are within classroom boundaries.

Flutter's native compilation ensures the 5-second check-in target is met, while its ecosystem supports ongoing development.

## System Design and Architecture

The system follows a client-server architecture:

- **Mobile Application**: Built with Flutter for Android and iOS, handling user authentication, facial capture, and location tracking.
- **Backend Server**: Node.js with Express manages APIs, data processing, and storage, using PostgreSQL for the database.
- **Database Schema**: Includes tables for users, courses, sessions, attendance, and facial feature vectors, stored securely.

### Key Features

- Real-time check-in with facial recognition.
- Geofencing to validate classroom presence.
- Instructor dashboard for session management and analytics.
- Student access to attendance history.

## Implementation Details

### Facial Recognition

- **Registration**: Students upload a facial image, processed by ML Kit to extract feature vectors, stored encrypted on the server.
- **Check-In**: The app captures an image, extracts features on-device, and sends them for server verification using cosine similarity, achieving up to 97.44% accuracy in similar systems (Face Recognition based Attendance Management System).

- **Performance**: On-device processing minimizes latency, meeting the 5-second target.

### Geofencing

- **Setup**: Classrooms are defined with latitude, longitude, and radius, stored on the server.
- **Verification**: During check-in, the app retrieves the current location via `geolocator` and validates it against the geofence, ensuring physical presence (Implementation Of Mobile Attendance Application Using Geo-Fence Technique).
- **Accuracy**: GPS is effective outdoors, with potential Wi-Fi enhancements for indoor settings.

### User Interface

- **Students**: Simple check-in screen with feedback (e.g., success/failure notifications) and attendance history view.
- **Instructors**: Dashboard with real-time attendance, filtering by course, date, or student, implemented via API polling or websockets.

### Security and Privacy

- **Encryption**: HTTPS for communications, encrypted storage of biometric data.
- **Authentication**: JWT for secure access.
- **Privacy**: On-device processing reduces data transmission, with compliance to regulations like GDPR where applicable.

## Performance Considerations

The system is optimized to meet the 5-second check-in goal:

- **Facial Recognition**: Inference times around 160ms ensure quick processing.
- **Geofencing**: Location checks are near-instantaneous.
- **Network**: Assumes reliable Wi-Fi, with offline capabilities planned for robustness.

## Challenges and Solutions

- **Facial Recognition Accuracy**: Variations in lighting or angles may affect performance. Solution: Collect multiple registration images and implement liveness detection.
- **Geofencing Accuracy**: Indoor GPS limitations may occur. Solution: Explore Wi-Fi or Bluetooth beacons for precise positioning.
- **Network Reliability**: Connectivity issues could disrupt check-ins. Solution: Offline mode with data syncing when connected.
- **Privacy Concerns**: Users may hesitate to share biometric data. Solution: Transparent policies and secure data handling.

## Comparative Analysis

Commercial solutions like Timeero and Truein rely on cloud-based facial recognition, introducing latency and privacy risks (AI Based Face Recognition Attendance System For Employees). This system prioritizes on-device processing, aligning with academic needs for data sovereignty and speed. Research supports high accuracy (97.44%) and efficiency in similar setups, reinforcing the approach (Face Recognition based Attendance Management System).

## Future Scope

Potential enhancements include:

- Liveness detection to prevent spoofing.
- Integration with university systems for seamless data flow.
- Multi-platform support beyond mobile devices.

## Conclusion

The requirement gathering process, supported by surveys, interviews, and research, has produced a robust design for a mobile-based attendance management system. Using Flutter, Google ML Kit, and geolocation services, the system addresses stakeholder needs for accuracy, efficiency, and usability. By tackling challenges like network reliability and privacy concerns, the project is well-positioned to deliver a transformative solution for academic attendance tracking.

## Technology Stack Table

| Component | Technology | Role |
|---|---|---|
| Mobile App | Flutter | Cross-platform development, UI, user interaction |
| Facial Recognition | Google ML Kit | On-device feature extraction and verification |
| Geofencing | geolocator plugin | Location tracking and boundary verification |
| Backend Server | Node.js, Express | API management, data processing, storage |
| Database | PostgreSQL | Store user data, feature vectors, attendance logs |

## Key Citations


- Face Recognition based Attendance Management System with High Accuracy
- Implementation of Mobile Attendance App Using Geo-Fence Technique
- Comparative Study of Flutter with Other Mobile App Frameworks
- Recognizing Faces in Android with TensorFlow Lite
- 7 Best Face Recognition Attendance Systems in 2024
- AI-Based Face Recognition Attendance System for Employees
- Google ML Kit for Flutter Face Detection
- Geolocator Plugin for Flutter Location Services