Chapter 1

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

The Smart Face Recognition Attendance System is designed to automate and enhance the process of attendance marking in educational institutions using advanced technologies such as facial recognition, GPS-based geofencing, and real-time data validation. This system eliminates the limitations of manual attendance and traditional biometric systems by introducing intelligent, secure, and location-verified automation through an Android application.

3.1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements of the Smart Face Recognition Attendance System. It provides a comprehensive overview of the system's behavior, design constraints, and interactions to ensure a mutual understanding between developers, users, and stakeholders. The document serves as a guideline for the development, testing, and validation of the system to ensure it meets its intended goals of automating attendance marking with high accuracy, security, and convenience.

3.1.2 Document Abbreviations / Definitions

Term / Abbreviation	**Definition**
AI	Artificial Intelligence
GPS	Global Positioning System
GUI	Graphical User Interface
SRS	Software Requirements Specification
API	Application Programming Interface
Liveness Detection	Verification process to ensure the user is physically present and not using a photo or video
Geofencing	Defining a virtual geographic boundary using GPS to restrict attendance marking to specific locations
SQLite	Lightweight local database management system used for storing attendance data
Firebase	Cloud-based platform used for authentication and real-time data storag

3.1.3 Target Audience

This SRS document is intended for the following audience:

- Developers – To design, implement, and test the application according to specified requirements.
- Project Supervisors / Guides – To review and ensure the system aligns with academic and technical objectives.
- Institutional Administrators and Faculty – To understand the system's functionality and monitor attendance reports.
- Students / End Users – To use the Android application for attendance marking.

3.1.4 Project Scope

Included:

Development of an Android-based smart attendance application for student use.
• Integration of live photo capture and AI-driven facial recognition technology.
• Implementation of GPS verification and geofencing features for enhanced security.
• Enforcement of time-window restrictions for attendance marking.
• Inclusion of notification alerts and auto-login functionalities.
• Utilization of cloud-based data storage and creation of real-time dashboards for faculty access.

Excluded:

- Integration with institutional ERP or LMS systems, reserved for future expansion.
- Attendance marking for offline or remote users located outside the campus premises.
- Deployment at multiple campuses or organizations, with the prototype focusing on a single campus setup.

3.1.5 Benefits

- **Accuracy:** AI-powered facial recognition minimizes manual errors.
- **Security:** Geofencing ensures attendance is marked only from authorized locations.
- **Efficiency:** Eliminates time-consuming manual attendance methods.
- **Transparency:** Cloud-based storage enables real-time monitoring by faculty.
- **Automation:** Auto-notifications reduce administrative follow-up efforts.
- **Scalability:** Modular design allows future integration with institutional systems.

3.1.6 References

1. Bramesh S. M., Arun P., and Deekshith H. R., *"Smart Attendance System with Facial Recognition and GPS Verification,"* **International Multidisciplinary Research Journal Reviews (IMRJR)**, Vol 2, Issue 8, Aug 2025. Available at: https://imrjr.com/papers/smart-attendance-system-with-facial-recognition-and-gps-verification/

2. H. S. Reddy, M. K. Gupta, A. S. Nayak, *"Geofenced Intelligent Attendance System Featuring Facial Recognition,"* **ResearchGate**, 2024. Available at: https://www.researchgate.net/publication/384677023_Geofenced_intelligent_attendance_system_featuring_facial_recognition

3. A. Rahman, K. Senthil, P. Rajesh, *"Mobile-Based Student Attendance System Using Geo-Fencing with Timing and Face Recognition,"* **ResearchGate**, 2023. Available at: https://www.researchgate.net/publication/361553012_Mobile_Based_Student_Attendance_System_Using_Geo-Fencing_With_Timing_and_Face_Recognition

3.2 Overall Description

This section provides an overview of the Smart Face Recognition Attendance System, including its relationship to existing systems, design perspective, product attributes, user characteristics, operational environment, constraints, assumptions, and dependencies. It establishes the context within which the system operates and describes the general factors that influence its design and functionality.

3.2.1 Identification of Pre-Existing Work

Existing attendance management systems primarily rely on manual or semi-automated methods such as:

- Manual registers – prone to errors, delays, and proxy attendance.
- RFID or ID swipe systems – require physical cards, which can be misplaced or shared.
- Fingerprint or biometric scanners – cause hygiene concerns and require physical contact.
- Basic face recognition systems – lack geolocation validation and time-based restrictions.

Although some recent systems utilize facial recognition, they fail to ensure live photo verification, location-based authentication, and time-window enforcement. The proposed system overcomes these shortcomings by integrating AI-driven facial recognition, GPS-based geofencing, liveness detection, and real-time notifications, thus providing a robust, contactless, and secure attendance mechanism.

3.2.2 Perspective on Product

The Smart Face Recognition Attendance System is an independent Android application designed for academic institutions. It captures live images of students, performs face recognition using AI, validates their presence within the geofenced location and scheduled time window, and records attendance securely to the cloud database.

From the user's perspective:

- Students use the app to mark their attendance by simply facing the mobile camera.
- Faculty can view real-time attendance data and analytics through a dashboard.
- Administrators can monitor records and generate reports from the centralized cloud system.

This system follows a modular and scalable architecture, allowing easy future integration with institutional ERP systems or expansion to multiple campuses.

3.2.3 Product Attributes

The system possesses the following key attributes:

Attribute	Description
Automation	Automatically identifies students and records attendance without manual entry.
Accuracy	Uses AI-based face recognition and liveness detection to ensure precise identification.
Security	Enforces GPS-based geofencing and time restrictions to prevent misuse.
Scalability	Designed with modular architecture for future integration and upgrades.
Usability	User-friendly Android interface with simple, intuitive navigation.
Portability	Operates on standard Android devices with camera and GPS capabilities.
Reliability	Real-time cloud storage ensures data integrity and availability.

3.2.4 End User Characteristics

User Type	Description / Characteristics
Students	Primary users; use the mobile app to mark attendance within authorized geofence areas during scheduled times. Should have basic knowledge of operating Android smartphones.
Faculty Members	Secondary users; access attendance data, view reports, and monitor student participation. Should be familiar with using web dashboards or mobile admin panels.
Administrators / Management	Manage system configuration, oversee multiple classes, and analyze institutional attendance records. Should have basic IT literacy.

3.2.5 Operating Environment

- Hardware Requirements :
  - Android smartphone with camera and GPS sensor (minimum Android 9 or higher).
  - Internet connectivity (Wi-Fi or mobile data).
  - For admin/faculty: computer or mobile device for dashboard access.
- Software Requirements :
  - Android Studio / VS Code for development and testing.
  - Python 3.11+ for backend logic (face recognition, database operations).
  - Libraries: OpenCV, face_recognition, NumPy, Pandas, TensorFlow Lite.
  - Cloud Database: Firebase for authentication and data storage.
  - SQLite for local temporary storage.
- Operating Systems Supported :
  - Android (for mobile application)
  - Windows, Linux, or macOS (for development and faculty access)

3.2.6 Constraints in Design and Implementation

1. Hardware Limitations: Accuracy depends on camera quality and lighting conditions during image capture.
2. Connectivity Issues: Real-time attendance requires stable internet for cloud synchronization.
3. GPS Accuracy: Geolocation precision may vary depending on network strength and device quality.
4. Privacy Concerns: Must comply with data protection policies for storing biometric and location data.
5. Performance: Facial recognition models must balance accuracy and response time for real-time processing.
6. Platform Limitation: Initial deployment restricted to Android platform only.
7. Environmental Conditions: Variations in lighting, camera angle, and face orientation may affect recognition accuracy.

3.2.7 Assumptions and Dependencies

Assumptions:

- Users have access to Android smartphones with camera and GPS features.
- Internet connectivity is available during attendance marking.
- Students are physically present within the geofenced location during class time.
- The institution provides accurate schedule and location details for each session.

Dependencies:

- System depends on external APIs such as Android Location Services and Firebase Cloud Storage.
- Performance depends on OpenCV and TensorFlow Lite for face recognition accuracy.
- Notifications rely on Firebase Cloud Messaging (FCM) for real-time updates.
- Database availability and reliability depend on Firebase’s uptime and connectivity.

3.3 Product Functionality

The Smart Face Recognition Attendance System is organized into modules that represent different user roles and functionalities. Each module interacts with the central database and cloud backend to ensure secure, accurate, and real-time attendance management. The system primarily consists of the Admin Module, Teacher Module, and Student Module.

3.3.1 Admin Module

Description:
The Admin module provides the highest level of access and control. Administrators manage user accounts, monitor system performance, and oversee attendance records across departments or classes. They can generate analytics, configure geofencing parameters, and ensure smooth operation of the system.

Functions:

1. User Management:
   - Register, update, or remove student and teacher accounts.
   - Assign roles and permissions.
2. Class and Schedule Management:
   - Define class timings and attendance windows.
   - Configure geofencing locations for attendance validation.
3. Monitoring and Analytics:
   - View consolidated attendance reports.
   - Generate performance insights (e.g., average attendance percentage, absentee trends).
4. System Configuration:
   - Manage database connections, cloud settings (Firebase).
   - Control notification templates and message content.
5. Security Management:
   - Authorize access privileges.
   - Maintain system logs for audit and tracking.

3.3.2 Teacher Module

Description:
The Teacher module enables faculty members to manage and monitor attendance for their respective classes. Teachers can view real-time data, verify attendance, and download reports for recordkeeping or submission.

Functions:

1. Attendance Monitoring:
   - View live attendance data from students during the session.

   - Verify student faces and ensure proper recognition.
2. Class Configuration:
   - Schedule attendance sessions for specific subjects and times.
   - Set or modify attendance windows if required.
3. Attendance Reports:
   - Generate daily, weekly, or monthly attendance reports.
   - Export attendance data to Excel or CSV format.
4. Notifications:
   - Send reminders or absence alerts to students via push notifications.
5. Performance Tracking:
   - Analyze attendance trends and identify consistently absent students.

3.3.3 Student Module

Description:
The Student module provides the front-end interface for students to mark their attendance using the Android application. The module validates the user's identity, location, and timing before confirming attendance.

Functions:

1. Login / Authentication:
   - Login using registered credentials (email/password or auto-login).
2. Face Recognition Attendance:
   - Capture a live image using the device camera.
   - Verify face using AI-based recognition and liveness detection.
3. Geofencing Validation:
   - Ensure attendance is marked within the designated GPS boundary.
4. Time Window Check:
   - Allow attendance only during the pre-specified class schedule.
5. Notifications:
   - Receive alerts on attendance submission or absence.
   - View attendance summary and status history.

3.4 External Interface Requirements

This section defines how the system interacts with users, hardware, and software components.

3.4.1 User Interface (UI)

- Mobile Application (Student Interface):
  - Simple and intuitive Android interface.
  - Login screen, attendance capture screen, and confirmation popup.
  - Displays real-time status (present/absent) and session details.
- Web or App Dashboard (Faculty/Admin Interface):
  - Dashboard with charts and attendance analytics.
  - List of students with attendance percentages.
  - Export and print options for reports.
- GUI Framework: Developed using Tkinter for desktop or Android XML for mobile interface.

3.4.2 Hardware Interface

- Camera:
  Required for capturing live student images for facial recognition.
- GPS Sensor:
  Used to determine user location for geofencing validation.
- Internet Connection:
  Required for real-time data synchronization with Firebase cloud.
- Device Requirements:
  - Android device with a minimum of 3 GB RAM and camera support.
  - For teachers/admins, a desktop or laptop for accessing analytics.

3.4.3 Software Interface

- Operating Systems:
  Android (for mobile app), Windows/Linux/macOS (for admin/faculty dashboards).
- APIs and Libraries:
  - OpenCV 4.7+ – Image processing and face detection.
  - TensorFlow Lite / MobileFaceNet – Facial recognition model.

  - Firebase – Authentication, database, and notification services.
  - SQLite 3+ – Local data storage and caching.
- Integration Points:
  - Firebase Cloud Messaging (for notifications).
  - Android Location API (for GPS and geofencing).

3.4.4 Communication Interface

- Cloud Communication:
  The app communicates with Firebase Realtime Database for storing attendance data, logs, and user information.
- Data Format:
  JSON is used for structured data exchange between the app and the cloud.
- Network Protocols:
  HTTPS protocol ensures secure data transfer between client and server.

3.5 Other Non-Functional Requirements

This section outlines the performance, safety, and software quality attributes that define how the Smart Face Recognition Attendance System must behave under various conditions. These requirements ensure that the system is secure, efficient, reliable, and user-friendly.

3.5.1 Performance Requirements

- The system must be able to detect and recognize a face within 2–3 seconds under normal lighting conditions.
- The app must support concurrent attendance marking for up to 50 students per session without performance lag.
- The face recognition accuracy rate should be at least 95% with clear lighting and frontal face capture.
- Database synchronization with Firebase should complete within 2 seconds for stable network connections.
- The system should operate smoothly on Android devices with at least 3 GB RAM and a stable internet connection (Wi-Fi or 4G).
- Notification delivery delay should not exceed 10 seconds after attendance is submitted.

3.5.2 Safety Requirements

- The system shall protect sensitive biometric data (facial images) by ensuring they are not stored locally after processing.
- All communication between the app and cloud database shall use HTTPS encryption to prevent data interception.
- Regular database backups shall be maintained in the cloud to prevent data loss.
- Only authorized users (admin, teachers, or registered students) shall have access to system modules.
- The app must handle unexpected crashes or network failures gracefully, with automatic reconnection or user prompts.
- All data shall comply with institutional privacy policies and standard data protection principles.

3.5.3 Software Quality Attributes

These attributes define the overall quality expectations of the Smart Face Recognition Attendance System.

3.5.3.1 Reliability

- The system shall maintain a 99% uptime during normal operation.
- Attendance data must be stored securely even in case of temporary network loss (local cache → sync to cloud).
- The application shall recover automatically after unexpected interruptions such as app closure or internet failure.

3.5.3.2 Usability

- The user interface must be simple, intuitive, and require minimal training.
- Icons and labels shall be self-explanatory for both students and teachers.
- Attendance status (Present/Absent) should be visually displayed immediately after submission.
- System navigation must not exceed three clicks for any main action (e.g., mark attendance, view report).

3.5.3.3 Efficiency

- System resources (CPU, memory, network) must be used optimally to ensure smooth app performance.
- The app must consume less than 100 MB of storage space on installation.
- Local caching and minimal API calls shall reduce data usage and network overhead.

3.5.3.4 Scalability

- The system architecture must allow future expansion to multiple campuses or institutions.
- The backend cloud database (Firebase) should support additional user growth without performance degradation.
- The face recognition model can be updated or replaced without redesigning the entire application.

3.5.3.5 Maintainability

- Code must be modular and well-documented to allow easy updates and debugging.
- New features (e.g., ERP integration, new recognition models) should be added with minimal impact on existing modules.
- Regular updates shall include bug fixes and performance improvements.

3.5.3.6 Security

- User credentials and biometric data must be encrypted before transmission or storage.
- Role-based authentication (Admin, Teacher, Student) must restrict access to sensitive features.
- The system must automatically log out inactive users after a defined period of inactivity.
- Firebase Authentication shall manage secure logins and prevent unauthorized access.

3.5.3.7 Portability

- The application must be compatible with Android versions 9.0 (Pie) and above.
- The backend and recognition modules should be deployable on different operating systems (Windows/Linux/macOS) without major changes.
- The app should adapt to various screen sizes (phones, tablets) with responsive design.

3.5.3.8 Interoperability

- The system shall integrate with other platforms such as ERP or LMS in future upgrades through RESTful APIs.
- Data formats (JSON) should remain compatible with third-party tools for reporting or analytics.
- The database structure should support export/import with CSV or Excel files for easy exchange.

3.5.3.9 Performance

- The app shall remain responsive under heavy load and maintain smooth UI performance.
- Real-time facial recognition should not exceed 3 seconds per detection.
- System latency between recognition and attendance logging must be below 2 seconds.

3.5.3.10 Flexibility

- The system design must support updates to attendance policies, class timings, or recognition parameters without code changes.
- New modules (like leave tracking or facial dataset management) should be easily integrated.
- The UI should allow customization (theme, notification settings) to suit institutional branding.s

3.6 Specific Requirements

This section describes the specific operating conditions, hardware, and software components required for the successful deployment and execution of the Smart Face Recognition Attendance System. These requirements define the technical environment in which the system will function effectively.

3.6.1 Operating Environment

The Smart Face Recognition Attendance System operates in a client-server environment, where the Android mobile application functions as the client, and the Firebase cloud platform acts as the backend server.

The system requires proper hardware and software components to ensure smooth functioning, accurate face recognition, and secure cloud synchronization.

3.6.1.1 Hardware Requirements

For Students (Mobile Application):

- Device Type: Android smartphone
- Processor: Minimum Octa-core 1.8 GHz or higher
- RAM: Minimum 3 GB
- Internal Storage: Minimum 100 MB free space for app installation
- Camera: Minimum 5 MP (front camera) with autofocus capability
- GPS Sensor: Mandatory for location-based attendance validation
- Internet Connectivity: Wi-Fi or 4G/5G mobile data
- Battery: Minimum 3000 mAh (to support extended app usage during sessions)

For Teachers and Administrators:

- Device Type: Desktop computer or laptop (for dashboard access)
- Processor: Intel Core i3 (or equivalent) or higher
- RAM: Minimum 4 GB
- Storage: Minimum 500 MB free space
- Display: Minimum resolution of 1366×768
- Internet Connection: Stable broadband or Wi-Fi connection

3.6.1.2 Software Requirements

For Development Environment:

- Programming Languages:
    - Python 3.11+ (for backend logic and face recognition)
    - Java/Kotlin (for Android application development)
- Frameworks / Libraries:
    - OpenCV 4.7+ – for image capture and processing
    - face_recognition 1.3+ – for facial detection and recognition

    - TensorFlow Lite / MobileFaceNet – for AI-based recognition model
    - NumPy, Pandas – for data handling and report generation
    - Tkinter – for GUI (if desktop prototype is used)
- Database:
    - SQLite 3+ – for local data caching
    - Firebase Realtime Database – for cloud data storage and synchronization
    - Firebase Authentication – for user login and role management
- Development Tools / IDE
    - Android Studio (for Android app design and testing)
    - Visual Studio Code (for backend and integration development)
- Operating Systems Supported:
    - Android 9.0 (Pie) or above – for mobile app users
    - Windows 10/11, Linux, or macOS – for admin/faculty dashboard or development
- Additional Software / APIs:
    - Android Location API – for geofencing and GPS tracking
    - Firebase Cloud Messaging (FCM) – for push notifications
    - JSON – for structured data exchange between app and cloud

3.7 Delivery Plan

The Delivery Plan outlines the major stages and timeline for the development, testing, and deployment of the Smart Face Recognition Attendance System. It defines the schedule of activities, expected deliverables, and milestones to ensure systematic and timely completion of the project.

3.7.1 Development Phases

The project will be developed and delivered in six key phases, each focusing on specific objectives and outputs.

Phase	Description	Deliverables	Duration
Phase 1	Requirement Analysis & System Design – Gathering user needs, analyzing existing	SRS Document, System Architecture Diagram	1 Week

	systems, and designing the system architecture and database model.		
Phase 2	UI/UX Design and Database Setup – Designing the Android interface and implementing SQLite for local storage.	Android UI Screens, SQLite Database Schema	1 Week
Phase 3	Face Recognition Module Implementation – Integrating OpenCV, TensorFlow Lite, and face_recognition libraries to develop and test recognition algorithms.	Functional Face Recognition Component	3 Weeks
Phase 4	Attendance Logging and Notification Module – Developing attendance marking, time-window validation, and notification system using Firebase.	Attendance Module with Cloud Sync and Alerts	2 Weeks
Phase 5	Testing and Debugging – Performing unit, integration, and user acceptance testing to ensure reliability, performance, and accuracy.	Test Report, Debugged Final Version	1 Week
Phase 6	Deployment and Documentation – Final deployment on Android devices, preparation of user manual, and submission of final project documentation.	Final App, Project Report, User Guide	1 Week

3.7.2 Deliverables Summary

At the end of the project, the following deliverables will be provided:

1. Functional Android Application for real-time attendance marking using face recognition and GPS validation.
2. System Documentation, including:
   - Software Requirements Specification (SRS)
   - Design Documents (UML, DFD, ER Diagrams)
   - Testing Reports
   - User Manual
3. Project Presentation and Demonstration showcasing live attendance marking and cloud synchronization.
4. Final Deployment Package, including source code, database files, and Firebase configuration.

3.7.3 Delivery Schedule

Activity	Start Date	End Date	Deliverable
Requirement Analysis & Design	Week 1	Week 1	SRS Document
UI/UX and Database Setup	Week 2	Week 2	Interface Prototype
Face Recognition Module	Week 3	Week 5	Recognition Component
Attendance & Notification Module	Week 6	Week 7	Working Attendance App
Testing & Debugging	Week 8	Week 8	Verified System
Final Deployment & Report Submission	Week 9	Week 9	Final Project Package

3.7.4 Delivery Method

- The final deliverable will be submitted as an Android APK file and project report document.
- Source code and related assets will be provided via a GitHub repository or zip archive.
- The project demonstration will be conducted on campus devices using the live application.
- Final documentation will be submitted in both digital (PDF) and printed formats.

3.7.5 Post-Delivery Maintenance

After project delivery:

- Minor bug fixes and optimizations will be carried out based on faculty feedback.
- Future upgrades may include integration with ERP systems, multi-campus support, and advanced analytics dashboards.