# Functional Requirements

These define **what** the system should do — the actual features and interactions.

| **No.** | **Functional Requirement** |
| --- | --- |
| 1 | **User Authentication:** Users (students, lecturers, IT admins) must log in securely using credentials (username/password or biometric authentication). |
| 2 | **Facial Recognition for Attendance:** Students must be able to mark attendance using facial recognition technology. |
| 3 | **Geofencing Verification:** The system must confirm students' physical presence inside a defined classroom boundary before accepting attendance. |
| 4 | **Attendance Session Management:** Lecturers must be able to initiate, open, and close attendance sessions. |
| 5 | **Real-time Attendance Recording:** System must immediately record and validate attendance once facial recognition and geolocation match. |
| 6 | **Offline Attendance Logging:** App must allow local (offline) attendance logging and sync when the network is available. |
| 7 | **Error Dispute Mechanism:** Students should be able to dispute incorrect absences through the mobile application. |
| 8 | **Notification System:** Students and lecturers must receive real-time notifications (attendance confirmation, errors, schedule changes). |
| 9 | **User Dashboards:** Students view attendance history; Lecturers view and manage class attendance records; Admins monitor overall system performance. |
| 10 | **Role-based Access Control:** Different features are available to students, lecturers, IT admins, and academic administrators based on their roles. |
| 11 | **Admin Controls:** IT administrators must manage system configurations, troubleshoot, and update user records. |
| 12 | **Data Analytics and Reporting:** System should generate attendance analytics and visualizations (graphs by class, semester, etc.). |
| 13 | **LMS Integration:** Attendance system should integrate with Learning Management Systems (LMS) for syncing records. |
| 14 | **Manual Override Option:** Lecturers must have an option to manually mark attendance in exceptional cases (e.g., device failures). |
| 15 | **Multiple Authentication Modes:** In addition to facial recognition, backup options like QR-code check-in or fingerprint scan should be available. |
| 16 | **Dynamic Geofence Update:** Lecturers must be able to create temporary class venues dynamically in the system. |
| 17 | **Student Notifications for Class Venue Change:** Students must be notified if a lecture's location changes. |
| 18 | **Survey Feedback Submission:** Students and lecturers should provide feedback within the app about user experience and technical issues. |

# Non-Functional Requirements

These define **how** the system performs — quality attributes like performance, security, usability.

| **No.** | **Non-Functional Requirement** |  |
| --- | --- | --- |
| 1 | **Performance:** Facial recognition must verify students within **5 seconds**. Geofencing must validate within **2 meters** accuracy. |  |
| 2 | **Availability:** System should have 99% uptime during academic hours (6:00 AM to 7:00 PM). |  |
| 3 | **Scalability:** The system must support **thousands of concurrent check-ins** during peak times (start of lectures). |  |
| 4 | **Security:** Facial data must be encrypted both **in transit** and **at rest**; secure APIs must be used for data communication. |  |
| 5 | **Privacy Compliance:** App must comply with **GDPR**, **local data protection laws**, and require explicit user consent for biometric processing. |  |
| 6 | **Battery Efficiency:** Mobile application must minimize GPS polling to avoid draining the student device’s battery. |  |
| 7 | **Offline Support:** Attendance data should be cached offline and synced automatically when the device reconnects to the internet. |  |
| 8 | **Cross-Platform Compatibility:** Mobile app must work seamlessly on **both Android and iOS** devices with a consistent UI/UX. |  |
| 9 | **Maintainability:** System codebase must follow modular design principles to facilitate easy updates and bug fixes. |  |
| 10 | **Extensibility:** Architecture must allow adding new biometric methods (e.g., fingerprint, voice) without redesigning the core system. |  |
| 11 | **Fault Tolerance:** In the event of app or server failure, system must queue attendance logs for retrying when service is restored. |  |
| 12 | **Auditability:** System must log all administrative and user activities for traceability and audit purposes. |  |
| 13 | **Data Retention Policy:** Biometric templates (not raw images) must be stored, with automatic deletion after a specified retention period (e.g., after semester ends). |  |
| 14 | **Usability:** Students and lecturers must complete attendance tasks with minimal steps (target: 3-click rule). |  |
| 15 | **Responsiveness:** Mobile app and web admin dashboards must respond in less than 1 second for basic operations. |  |
| 16 | **Real-Time Data Synchronization:** Attendance status should update across all devices instantly upon marking. |  |
| 17 | **Disaster Recovery:** Backups must be maintained daily with a maximum data loss tolerance (RPO) of 24 hours. |  |
| 18 | **Accessibility:** The mobile application must be usable by users with disabilities (basic WCAG compliance). |  |
| 19 | **Cost Efficiency:** Use open-source technologies and scalable cloud services to optimize operational costs. |  |