Software Requirements Specification (SRS)

Introduction / Scope:

This SRS documents functional and non-functional requirements for the School Management Application. The system will support typical school administrative tasks: student records, attendance, timetabling, exam/marks management, fee collection, and provide role-based access for administrators, principals, teachers, students, parents and accountants.

User Roles & Descriptions

Admin — System configuration, user management, global policies.

Principal — Oversight and approvals, high-level reporting.

Teacher — Class-level operations (attendance, marks), content sharing, parent messaging.

Student — Views only: timetable, attendance, marks and resources.

Parent — Views child data, receives notifications, views/receives fee receipts.

Accountant — Fee structure manager, payment recording, basic finance reports.

Functional Requirements:

Student Management

Add, edit, archive student profile (personal details, DOB, enrollment number, guardian info).

Assign student to class and section.

Upload and store documents (birth certificate, TC, ID proofs).

Search and filter student records by class, name, enrollment ID.

Bulk import student data (CSV).

Attendance Management

Mark daily attendance at class or individual student level.

Bulk import/export attendance via CSV.

Generate attendance percentage reports and history per student.

Send absence notifications to parent (email/SMS/push) when marked absent or below threshold.

Teacher-level and admin-level attendance views (teacher sees own classes; admin sees all).

Timetable Management

Create weekly timetable per class/section.

Assign teachers and subjects to time slots.

Detect conflicts (teacher double-booked, class clashes).

Publish and notify timetable changes to students/parents/teachers.

Exam & Marks Management

Create exam entries and schedules per class.

Define max marks and optional weightages.

Enter marks per student/subject (manual or bulk import).

Calculate total marks, grades and generate report cards.

Maintain audit trail for mark changes.

Fees Management

Define fee structure per class and optional components (tuition, transport).

Generate invoices with unique invoice numbers.

Record payments (partial/full), issue receipts.

View outstanding balances and send fee reminders.

Export payment reports for accountant.

Teacher Module

Access class lists, student lookup and profiles.

Mark attendance and enter marks.

Post assignments, class notes and announcements.

Initiate parent-teacher messages/meeting requests.

Parent Portal

View child’s profile, attendance summary, marks and timetable.

Receive fee reminders and download receipts.

View announcements and download shared study material.

Update guardian contact info for notifications.

Non-Functional Requirements

Performance

Typical page load times under normal school load should be snappy (goal: <2–3s for common pages).

Backend APIs should handle concurrent usage supporting a typical school (hundreds to low thousands of active users).

Security

Role-Based Access Control (RBAC) with least-privilege.

Authentication: secure password storage (hashed), session management, optional 2FA for sensitive roles.

Encrypted transport (HTTPS/TLS). Data-at-rest protections for sensitive documents where needed.

Scalability

Architected to allow horizontal scaling of backend and read-replicas for reporting.

Static assets and uploaded documents served via CDN/object storage (S3-like) as load grows.

Usability

Simple, mobile-responsive UI; minimal steps for frequent tasks (attendance, marks entry, invoice generation).

Clear labels, tooltips and confirmation dialogs for destructive actions.

Availability

Target availability during school operational hours; aim for >99% excluding planned maintenance.

Daily/nightly backups and a tested restore procedure.

System Rules & Constraints

Data retention: Student records and financial receipts must be retained for X years (define per school policy).

Role restrictions: Only accountant/admin can modify fee structures; only teachers assigned to a class can mark that class's attendance.

Unique constraints: Enrollment numbers, invoice numbers and user email addresses must be unique.

Validation rules: Required fields (student name, DOB, guardian contact), numeric limits for marks (0..max_mark).

Auditability: All edits to marks, fees and student critical data must keep an audit trail (who, when, old value, new value).

Offline fallback: Basic ability for teachers to import attendance/marks via CSV if live entry isn't possible.
