# School Information System (SIS) - Business Requirements Document

## 1. Executive Summary

This document outlines the business requirements for the development of a next-generation School Information System (SIS) designed to serve the needs of administrators, teachers, specialists, students, and parents. The SIS aims to integrate core operational functions—such as student information, scheduling, attendance, grading, and communication—into a unified, intuitive platform. The project is divided into multiple phases (A–E) to ensure structured implementation, reduce development risk, and allow for early user feedback and incremental improvements.

## 2. Background & Current Challenges

Current SIS solutions in use across schools, particularly private K–8 institutions, often fail to provide seamless integration between administrative, instructional, and parental engagement functions. Teachers are frequently required to use multiple disconnected tools, resulting in inefficiencies, duplicated data entry, and poor year-to-year continuity. Parents often receive fragmented and delayed information about their children’s progress, while administrators lack real-time insights into school-wide performance metrics.  
  
From both teacher and parent perspectives, current systems present significant usability issues. Teachers report that the platforms are cumbersome and not tailored to the realities of daily classroom management, while parents find the available portals unhelpful in tracking progress or addressing concerns in a timely manner. Administrators struggle with data silos, limited reporting functionality, and lack of analytics to inform staffing or curriculum decisions.

## 3. Project Objectives & Goals

The primary objective of this project is to deliver a best-in-class SIS platform that:  
• Enhances collaboration between teachers, parents, and administrators.  
• Centralizes key academic and operational data into one secure system.  
• Provides intuitive, role-based interfaces tailored for each user type.  
• Streamlines administrative tasks such as scheduling, grading, and reporting.  
• Offers powerful analytics for tracking student performance and teacher impact year over year.  
• Integrates specialists into the scheduling and reporting structure from the start.  
• Supports both structured (middle school) and flexible (grade school) scheduling models.  
• Lays the groundwork for AI-assisted recommendations in scheduling, workload balancing, and student support.

## 4. Scope & Phased Implementation (A–E)

The project will be executed in five structured phases to ensure manageable development cycles and incremental delivery of value:  
  
Phase A – Core Setup & Academic Structure:  
 • School creation and configuration (district support optional at MVP).  
 • Student, teacher, specialist, and admin onboarding.  
 • Core subject and special creation with time allocations.  
 • Gradebook structure with customizable grading scales.  
 • Attendance tracking (excused, unexcused, late, present, etc.).  
 • Scheduling framework for K–5 (teacher swap flexibility) and 6–8 (rotating classes).  
  
Phase B – Communication & Parent Portal:  
 • Parent and guardian account setup.  
 • Messaging and announcement system.  
 • Homework and assignment visibility.  
 • Performance dashboards for parents.  
  
Phase C – Advanced Analytics & Reporting:  
 • Teacher workload analysis by year and subject.  
 • Student performance tracking over multiple years.  
 • Comparative analytics for teacher impact.  
 • Curriculum pacing and engagement metrics.  
  
Phase D – Lesson Planning & Resource Management:  
 • Teacher lesson plan creation and storage.  
 • Integration of lesson plans into scheduling.  
 • Resource attachment and sharing.  
  
Phase E – Future Enhancements & AI Integration:  
 • AI-assisted scheduling.  
 • Predictive analytics for student success.  
 • Automated parent updates based on student performance triggers.

## 5. Stakeholder Analysis

Key stakeholders include:  
• Administrators – Require tools for configuration, oversight, analytics, and compliance.  
• Teachers – Need simple yet powerful tools for attendance, grading, and scheduling.  
• Specialists – Must coordinate schedules with classroom teachers while avoiding conflicts.  
• Parents/Guardians – Expect real-time access to student progress and easy communication.  
• Students – Will benefit from organized schedules, assignment tracking, and clear performance data.

# SIS – Phase A Business Requirements (Sections 6–11)

Version: 1.1 • Scope: Core SIS foundation (School, Students, Teachers, Specialists, Scheduling, Attendance, Gradebook)

## 6. Scheduling (Core + Specialists)

### 6.1 Core Scheduling

Grade School (K–5): Admin assigns specials (locked blocks with weekly minutes). Teachers arrange core subjects around specials; partner-teaching and student swaps allowed. Weekly minute constraints enforced, not daily. Middle School (6–8): Admin builds master schedule (rotations, rooms, teachers). Teachers may adjust non-core enrichment with admin approval.

### 6.2 Calendar UI Requirements

True calendar views (day/week/month). Time blocks have custom start/end, color coding by subject/teacher/grade. Conflict detection for teacher, student, and room overlaps. Views/filters: By Teacher, Grade, Student, Subject. Clicking a block can open attendance for that period. Future: lesson plans and attachments visible from the calendar.

## 7. Specialist Pull-Out Scheduling

Request/Approval: Specialists can request ANY time. Requests route to the student’s primary teacher for approval. All pull-outs inherently overlap class; teacher approval required. If denied, teacher selects a reason (list/custom) and can propose an alternative directly from the specialist’s available calendar. Specialists may host multiple students per slot up to a max capacity set by specialist or admin. Notifications for approve/deny/alternate proposals. Admin (principal/VP/dean/staff) can view/override; full history retained (timestamps, reasons, subjects impacted).

## 8. Attendance Management

Entry & Defaults: Teacher or admin/staff can mark attendance. If no status is entered for a student during the attendance window, default to Present. Statuses: Present, Absent–Excused, Absent–Unexcused, Tardy–Excused, Tardy–Unexcused (extendable per policy). Parents see status only (no internal notes). Attendance links to calendar (click class block → attendance sheet). Admin reports by student/class/teacher/date/reason.

## 9. Gradebook Management (with Flexible Groupings)

### 9.1 Category Groupings

Teachers/admin can define grading groupings (e.g., Homework, Quizzes, Tests, Projects, Independent Work, Participation). Per-class configuration with options:  
• Weight per category (default weight = 1; supports >1 or <1 per class policy).  
• Drop-lowest setting per category (optional).  
• Default category for new assignments (optional).  
• Reorder categories (affects display only).

### 9.2 Assignment Creation & Inline Category Flow

Two entry modes for maximum flexibility:  
A) Assignment-first: Teacher clicks “New Assignment” → enters Title, Description, Max Points, Due Date, Category, Weight (optional), Late Policy (inherit class default, override allowed) → Save. Students auto-linked via current roster.  
B) Category-first (rapid entry): Teacher clicks a Category chip (e.g., Homework) → “Quick Add Assignment” appears inline above the grid → enter Title (required) and optional Max Points/Weight → instantly opens the grade grid for that assignment to enter scores row-by-row.  
Inline Category Create: If the needed category doesn’t exist, teacher can create a new category on-the-fly from the same dialog without leaving the grid.

### 9.3 Grade Entry & Calculations

Spreadsheet-speed grid (keyboard navigation, pasting from clipboard). Status badges: Missing, Late, Excused, In Progress. Calculations update in real time with two toggles: (1) Include Missing as zero; (2) Exclude Missing. Late penalties auto-applied per rule (percent or points) with visual hint. Extra credit supported (assignment flagged as Extra Credit). Per-student comments allowed; parents see comment if marked ‘shareable’. All grade changes logged with timestamp, user, and optional reason (retain ≥3 years).

### 9.4 Visibility & Access

Teachers: only their classes. Admin: all classes. Parents/Students: only student’s own results; parents see category weights and assignment detail, but not internal teacher notes unless marked shareable.

## 10. Admin Role Hierarchy & Permissions

Principal: Full school access and settings. Vice Principal: Full access except certain confidential HR/discipline (configurable). Dean: Focus on conduct/attendance/interventions; can approve pull-outs. Staff (Admin): Nearly full access by default, but structured to easily restrict specific capabilities later (e.g., global settings). Future: custom granular permissions with role scopes.

## 11. Reporting & Analytics (Phase A Scope)

Admin: Grade distribution by class/subject/teacher; attendance trends by day/week/month; specialist pull-out frequency and impact; schedule coverage vs. weekly minute requirements. Teacher: Missing work list; category averages; per-student progress over current term. Parent: Report-card style view with category breakdown and assignment details; attendance summary.  
  
Foundational Metrics (Phase A):  
• Attendance completion time: p50 ≤ 2 minutes/class; p95 ≤ 4 minutes.  
• Grade entry throughput: 25 students in ≤ 5 minutes for a single assignment (p50).  
• Specialist approval turnaround: median ≤ 24 hours.  
• Schedule conflicts resolved prior to publish: ≥ 99%.  
• Parent weekly engagement: ≥ 70% log-in at least once/week by end of pilot.

# SIS – Phase B Business Requirements (Sections 6–11)

Version: 1.1 • Phase B: Curriculum & Lesson Planning • Audience: Product, Engineering, QA, School Leadership

## 6. Functional Requirements

### 6.1 Scope & Goals

Deliver best-in-class curriculum and lesson planning tightly integrated with Phase A (Core SIS foundation: classes, schedules, gradebook, attendance, specialists). Phase B introduces curriculum maps, unit plans, and lesson plans that directly reference Phase A entities so teachers never re-enter data. Outputs must feed forward into Phase C (Assessments) and Phase D (Communication) while preserving longitudinal continuity for Phase E (Analytics).

### 6.2 Planning Levels

• Curriculum Map (Year/Term): subject/grade pacing with units, standards targets, and milestones.  
• Unit Plan (Multi-week): objectives, standards alignment, assessments, resources, differentiation, checkpoints.  
• Lesson Plan (Daily/Block): objectives, activities with minutes, materials, accommodations, exit tickets, attachments, linked assignments and assessments.

### 6.3 Lesson Plan Structure

Each lesson supports: title; class/grade (from Phase A roster); subject; date or date range; duration; objectives; standards alignment; activities (sequenced with estimated minutes); materials/resources; differentiation (groups, accommodations, extensions); checks for understanding; homework; attachments/links; and visibility controls (teacher-only, team, admin, parent-friendly).

### 6.4 Templates & Reuse

• Built-in templates for K–5 and 6–8 (e.g., Gradual Release, Workshop, Inquiry).  
• School-custom templates (admin-managed) and teacher personal templates.  
• Copy/clone lessons and units across dates, classes, and years; bulk shift by calendar.  
• Version history with restore; draft/published states; lock after publish (admin can unlock).

### 6.5 Standards Alignment

• Attach multiple standards per lesson/unit; support state/national/local frameworks.  
• Standards library with search, tags, and aliases; admin can import sets.  
• Coverage tracking (percent of targeted standards planned vs taught), with links to associated assessments (Phase C).

### 6.6 Schedule Integration (Phase A Dependency)

• Lessons auto-link to Phase A calendar blocks by class/teacher; teachers can drag lessons onto blocks or auto-insert.  
• When a time block moves in the calendar, linked lessons can shift (single or batch). • Calendar block shows lesson summary and attachments; attendance can be taken from the block (Phase A).

### 6.7 Gradebook & Assessment Linkage (Phases A & C)

• From a lesson, teachers can create assignments (title, max points, category, weight, due date) that publish to the Phase A gradebook. • Lessons can also link to Phase C assessments; standards alignment flows through to item blueprints and mastery analytics.

### 6.8 Collaboration & Sharing

• Co-teacher editing; grade-level and subject team sharing; school libraries for exemplary units.  
• Comments and @mentions; suggestion mode with accept/resolve.  
• Admin read-only oversight; selective parent-friendly summaries.

### 6.9 Specialists & Differentiation (Phase A Dependency)

• Specialists can annotate lessons with intervention goals/activities for students they support. • Lesson view flags scheduled pull-outs (from Phase A specialist scheduling) so teachers can plan catch-up tasks.

### 6.10 Parent Visibility Controls (Feeds Phase D)

• Toggle ‘Parent-friendly summary’ at the lesson level. Internal fields (accommodations, internal notes) are hidden. • Weekly digest view aggregates lesson summaries per class for Phase D communication channels.

### 6.11 Search & Organization

• Global search by title, objective, standard, tag, resource type; filters by grade, subject, author, date range.  
• Tagging system for lessons/units/resources (e.g., SEL, STEM, Literacy). Favorites and pins for quick access.

## 7. Non-Functional Requirements

• Performance: autosave < 500ms; open lesson < 1.5s p95; copy 20 lessons < 5s.  
• Reliability: version history ≥ 3 years; conflict resolution for concurrent edits.  
• Accessibility: WCAG 2.1 AA; keyboard-first editing; high-contrast and large-text modes.  
• Security: role-based visibility; internal fields encrypted at rest; audit publish/lock events.  
• Data Integrity: lessons must reference valid Phase A class/teacher IDs; orphan prevention checks on copy/clone.

## 8. Data Requirements

Core Entities (referencing Phase A IDs):  
• CurriculumMap(map\_id, school\_id, subject, grade\_level, year, pacing\_targets, standards\_targets, created\_by, visibility)  
• UnitPlan(unit\_id, map\_id?, title, start\_date, end\_date, objectives, standards[], resources[], assessments[], tags[], owner, visibility, status, version)  
• LessonPlan(lesson\_id, class\_id [Phase A], teacher\_id [Phase A], date, duration, objectives, activities[], materials[], standards[], accommodations, checks, homework, attachments[], visibility, status, version)  
• LessonTemplate(template\_id, owner/school\_id, structure\_schema, name, visibility)  
• LessonAttachment(file\_id, lesson\_id, type, url/path, name, size, uploaded\_by, shareable)  
• ResourceLibrary(resource\_id, owner/school\_id, type, title, description, tags[], url/path, standards[], visibility)  
• LessonScheduleLink(link\_id, lesson\_id, calendar\_block\_id [Phase A])  
• LessonAssignmentLink(link\_id, lesson\_id, assignment\_id [Phase A])  
• Comment(comment\_id, target\_type, target\_id, body, author, mentions[], created\_at, resolved)  
• Standard(standard\_id, code, description, subject, grade\_band, jurisdiction, aliases[])  
• Tag(tag\_id, name, color, scope)  
  
Constraints & Rules:  
• Parent-friendly view is derived (not a raw field dump).   
• Standards references must resolve to managed sets.   
• Version bumps on publish and major edits; drafts stored as deltas.

## 9. Reporting & Analytics (Phase B)

• Standards Coverage: % planned vs targeted by term; % delivered (linked to calendar).   
• Pacing Adherence: planned vs actual timeline; slippage alerts; spillover indicators.   
• Resource Utilization: most used templates/resources; reuse rate across teachers/years.   
• Collaboration Metrics: co-authored lessons; comment/resolve cycles.   
• Parent Visibility: lessons with parent friendly summaries; weekly digest counts (feeds Phase D).

## 10. Workflow & Process Requirements

### 10.1 Year/Term Setup (Depends on Phase A)

Admin defines curriculum maps per subject/grade → targets standards and pacing → assigns exemplars/templates → teachers receive starting maps linked to their Phase A classes.

### 10.2 Unit & Lesson Authoring

Teacher selects template → drafts unit objectives and assessments → creates lessons (copy or new) → aligns standards → adds materials and activities with minute estimates → links lessons to calendar blocks (Phase A) → marks draft/publish.

### 10.3 Collaboration & Review

Co-teachers and team leads comment and suggest changes → teacher accepts/implements → version saved on publish → admin read-only oversight; can request changes via comments.

### 10.4 Gradebook & Assessment Linkage (Phases A & C)

From a lesson, teacher creates assignments that auto-appear in Phase A gradebook. Lessons can reference Phase C assessments; standards alignment supports mastery analytics in Phase E.

### 10.5 Parent-Friendly Publishing (Feeds Phase D)

Teacher toggles parent-friendly summary → system composes concise view (objectives, homework, shareable attachments) → published to parent portal and included in Phase D weekly digest.

## 11. Phase B Metrics & Acceptance

Acceptance Criteria:  
• A teacher can create a unit with ≥ 5 lessons, align standards, and link each lesson to a calendar block in ≤ 30 minutes.  
• Creating an assignment from a lesson appears in the Gradebook within 2 seconds.  
• Copying a week of lessons across two classes and shifting by one week completes in ≤ 10 seconds.  
• Parent-friendly summaries are visible in the portal and weekly digests for ≥ 80% of published lessons by week 6 of pilot.  
  
Success Metrics (Phase B):  
• Teacher time to author a daily lesson (median) ≤ 10 minutes using templates.  
• Reuse rate of lessons/templates ≥ 40% by end of term.  
• Standards coverage variance ≤ 10% from pacing targets by grade/subject.  
• Parent satisfaction (lesson clarity) ≥ 80% in monthly pulse survey.  
  
Dependencies Recap:  
• Requires Phase A classes, schedules, gradebook, and attendance to exist.   
• Feeds Phase C item blueprints and mastery analytics; feeds Phase D weekly digests; supports Phase E longitudinal analysis.

# Phase C – Advanced Grading, Assessment, and Analytics

(Dependencies: Phase A – Core Data Structures & Attendance, Phase B – Scheduling & Specialist Management)

## 1. Purpose & Goals

Transform the basic gradebook into a comprehensive, teacher-friendly, analytics-driven assessment system that:

• Simplifies grading for teachers.

• Allows online student assessment with auto-grading.

• Provides actionable insights for teachers, students, parents, and administrators.

• Integrates seamlessly with schedules and attendance to support whole-student performance tracking.

## 2. Core Functional Requirements

### 2.1 Assignment & Grade Management

Teachers can:  
- Create assignments directly in the gradebook or link to scheduled lessons (from Phase B).  
- Define categories (tests, quizzes, homework, participation, projects, etc.) with custom weights.  
- Apply rubrics for qualitative grading.  
- Bulk-create assignments across multiple classes.  
Assignments can:  
- Have due dates synced with the class schedule.  
- Be reused year-to-year.  
- Include attached resources (PDFs, videos, slides).  
- Allow late submissions with auto-applied penalties.

### 2.2 Student Assessment Delivery

Online student portal to:  
- View assignments.  
- Submit work directly.  
- Take quizzes/tests online.  
Assessment formats supported:  
- Multiple choice, true/false, short answer, fill-in-the-blank (auto-graded).  
- Long-form written responses and projects (manual grading).  
Auto-grading:  
- Configurable correct answer keys.  
- Partial credit support.  
- Instant scoring for objective items.  
Teacher override for any auto-graded score.

### 2.3 Analytics & Early Intervention

Real-time grade trend graphs per student, per class, per school.  
- Identify students at risk:  
 \* Grade drop alerts.  
 \* Missing assignments tracking.  
 \* Performance dips tied to attendance patterns.  
- Compare class averages to historical performance.  
- Aggregate reporting for:  
 \* Teachers (class-level trends).  
 \* Admin (school-level performance metrics).  
 \* Specialists (impact of pull-outs on academic progress).

### 2.4 Parent & Student Views

Parents:  
- See grades, missing work, attendance correlation.  
- Receive alerts for failing grades or missing work.  
Students:  
- See grades, teacher feedback, and test results.  
- Access review materials linked to missed concepts.

### 2.5 Permissions & Roles

Teacher: Full gradebook editing for own classes, feedback control.  
Admin (Principal, VP, Dean): View all gradebooks, run performance reports.  
Specialists: Limited view for students they serve, tied to their schedule.  
Parents/Students: Read-only with alerts.

### 2.6 Workflow Integrations

Sync due dates with Phase B schedule to prevent conflicts.  
Pull attendance from Phase A to overlay performance trends.  
Tie missing work alerts to future communication phase for auto-messaging.

## 3. Non-Functional Requirements

Performance: Auto-grading must process and update results instantly for classes up to 40 students.  
Scalability: Must support multi-school districts without performance loss.  
Security: FERPA-compliant; only authorized users can view/edit grades.

## 4. Dependencies

Requires completed Phase A for core student/teacher/class data and attendance.  
Requires completed Phase B for schedule alignment and specialist availability.

## 5. Success Metrics

90%+ of teachers using the online gradebook weekly.  
Parent login rates above 75% by mid-year.  
At least 50% of assessments using online delivery by year two.

# Phase D: Communication & Engagement - Business Requirements

## 1. Purpose & Scope

The purpose of Phase D is to design and implement a best-in-class communication and engagement module that connects administrators, teachers, parents, and students through a unified platform. This phase focuses on timely, secure, and effective communication channels that are integrated with core SIS data (student profiles, schedules, grades, attendance, and specialist appointments).

## 2. Dependencies

Phase D is dependent on Phases A–C for foundational data:  
- Phase A: Student, teacher, and specialist profiles; attendance and schedule data.  
- Phase B: Specialist scheduling and approval workflows.  
- Phase C: Grading, assessment results, and grouping structures.  
Without these phases, communication cannot be properly linked to context-specific information.

## 3. Communication Modes

The system must support multiple modes of communication:  
1. In-App Messaging – Real-time secure messaging between roles.  
2. Email Integration – Automated and manual email sending with message logs.  
3. Push Notifications – Mobile alerts for urgent updates or reminders.  
4. Calendar Reminders – Automated event and deadline notifications.  
5. Bulletin Boards – Public announcements visible to specified audiences.

## 4. Permissions & Roles

Each communication feature must be role-aware:  
- Administrators: Broadcast to all users, initiate group messages, send alerts.  
- Teachers: Message students, parents, and administrators; share resources.  
- Specialists: Coordinate pull-outs, request changes, share progress updates.  
- Parents: Communicate with teachers, specialists, and admins regarding their child.  
- Students: Communicate with teachers and specialists in an academic context.

## 5. Engagement Features

Best-in-class engagement features include:  
- Automated progress summaries sent to parents weekly or monthly.  
- Multilingual support for all communications.  
- Accessibility compliance (screen reader support, high-contrast mode).  
- Interactive event invitations with RSVP tracking.  
- Resource sharing (documents, videos, images) within messages.

## 6. Security & Privacy

All communication must comply with FERPA and relevant data privacy laws.  
Requirements:  
- End-to-end encryption for in-app messages.  
- Audit logging of all messages and file shares.  
- Role-based access control to restrict sensitive information.  
- Consent tracking for parent/student communication preferences.

## 7. Future-State Integrations

In later phases, communication data will integrate with analytics to:  
- Track engagement frequency and quality.  
- Identify communication gaps impacting student performance.  
- Correlate teacher-parent communication with grade and attendance trends.

# Phase E – Reporting, Analytics, and Insights (Business Requirements)

Version: 1.0 • Audience: Product, Engineering, Data, QA, School/District Leadership

Dependencies: Phase A (Core SIS, Scheduling, Attendance, Gradebook, Specialists), Phase B (Curriculum & Lessons), Phase C (Assessments & Auto‑Grading), Phase D (Communication & Engagement)

Purpose: Deliver best‑in‑class analytics that help leaders, teachers, specialists, parents, and students act early—without creating busywork or ambiguity. All insights must be explainable, auditable, and tied to concrete actions.

## 6. Functional Requirements

### 6.1 Executive & School Dashboards

• District/School Health: attendance rate (daily/weekly/term), chronic absence %, grade distributions, assessment mastery, missing work counts, discipline incidents, specialist utilization, communication engagement (Phase D).   
• Drill‑downs: district → school → grade level → teacher → class → student.   
• Time Context: this week, MTD, term, year, prior year, multi‑year trends.   
• Equity Slices: cohort/subgroup filters (ELL, IEP/504, grade band, program).

### 6.2 Teacher Analytics & Workload

• Class Heatmaps: standards mastery, missing work, late penalties impact, attendance‑achievement correlation.   
• Workload View: grading volume, time-to-grade, feedback coverage, communication follow‑through (Phase D).   
• Action Lists: prioritized list of students needing attention with reasons and suggested actions.

### 6.3 Student 360 & Intervention Tracker

• Student 360: unified view of grades, assessment history, attendance, schedule, pull‑outs (Phase A), accommodations, lesson links (Phase B), communications (Phase D).   
• Interventions: create/assign/track interventions (tutoring, check‑ins, study plans).   
• Outcomes: link interventions to subsequent grade/attendance changes; calculate impact deltas.

### 6.4 Standards Mastery & Curriculum Analytics

• Coverage vs Mastery: planned standards (Phase B) vs assessed mastery (Phase C).   
• Pacing: schedule alignment (Phase A/B) vs delivery and mastery lag indicators.   
• Resource Effectiveness: correlate lesson resources/templates (Phase B) with mastery outcomes (Phase C).

### 6.5 Assessments & Gradebook Analytics

• Item Analysis: difficulty, distractor performance, rubric criterion trends.   
• Gradebook Mix: category weight effects, extra credit impact, missing/late sensitivity.   
• Retake Policy Effects: compare highest/latest/average policy outcomes across classes.

### 6.6 Attendance & Engagement Analytics

• Chronic Absence Insights: flag at‑risk thresholds; correlate with grades and assessment dips.   
• Engagement Signals: read receipts, message reply latency, portal logins (Phase D) as early warning signals.   
• Specialist Pull‑Out Impact: attendance during sessions, frequency, and academic correlation.

### 6.7 Alerts, Subscriptions, and Nudges

• Threshold‑Based Alerts: grade drop > X points, attendance < Y%, missing > N assignments.   
• Smart Subscriptions: principals auto‑receive weekly school summaries; teachers get class snapshots; parents get student digests.   
• Explainability: every alert shows contributing data and suggested next actions (message parent, assign intervention, schedule check‑in).

### 6.8 Data Export & APIs

• Self‑Service Exports: CSV/Excel/PDF for any report with applied filters and metadata footers.   
• Programmatic Access: secure REST/GraphQL endpoints for dashboards and aggregates (role‑scoped).   
• Scheduled Delivery: email/SFTP cadence for district reporting packs.

## 7. Non‑Functional Requirements

• Accuracy & Definitions: a governed metrics catalog (single source of truth) with formulae, inclusions/exclusions, and update frequency.   
• Performance: dashboard initial render ≤ 3s p95; heavy reports ≤ 7s p95; exports ≤ 60s p95.   
• Freshness: near‑real‑time for class‑level views (≤ 5 min); hourly for school‑level; daily for district packs.   
• Security: FERPA‑compliant RBAC; row‑ and column‑level security (RLS/CLS); PII minimization in exports; signed URLs.   
• Governance: audit lineage from source event to metric; change control on metric formulas; versioned dashboards.   
• Accessibility: WCAG 2.1 AA; keyboard shortcuts; high‑contrast and screen reader friendly tables and charts.   
• Reliability: 99.5% monthly availability; degraded‑mode summaries when source feeds are delayed.

## 8. Data Requirements

### 8.1 Semantic Layer (Business‑Friendly Model)

• Metrics Dictionary: attendance rate, chronic absence, average grade, mastery %, missing count, late %, intervention impact delta, communication engagement, specialist utilization.   
• Conformed Dimensions: Student, Teacher, Class, School, Term, Subject, Standard, Calendar, Cohort, Program.

### 8.2 Data Model (Warehouse Schema)

Facts (event/measure tables):   
• fact\_grades (student\_id, class\_id, assignment\_id, points\_earned, points\_possible, late\_flag, missing\_flag, timestamp)   
• fact\_assessments (attempt\_id, assessment\_id, score, item\_stats, rubric\_scores, duration, retake\_seq, timestamp)   
• fact\_attendance (student\_id, date, class\_id?, status, tardy\_minutes, notes\_flag, recorded\_by, timestamp)   
• fact\_schedule (block\_id, class\_id, teacher\_id, student\_count, minutes\_planned, minutes\_delivered)   
• fact\_pullouts (session\_id, specialist\_id, student\_id, duration, reason, approved\_by, timestamp)   
• fact\_communications (thread\_id, sender\_role, recipient\_role, read\_time, reply\_latency, attachments\_flag)   
• fact\_interventions (plan\_id, student\_id, type, start/end, dosage, owner, outcome\_metrics, status)   
  
Dimensions:   
• dim\_student (SCD2: enrollment status, cohorts, programs)   
• dim\_teacher (SCD2: assignments, departments)   
• dim\_class (course, grade\_level, category\_weights)   
• dim\_school (school\_type, calendar, bell\_schedules)   
• dim\_subject, dim\_standard, dim\_calendar (date, week, term, year), dim\_cohort, dim\_program.

### 8.3 Data Lineage & Quality

• Lineage: trace metric → aggregate → facts → source event/API.   
• Quality Checks: schema conformance, null/duplicate detection, range checks (e.g., % within 0–100), reconciliation to SIS counts.   
• Backfills: safe historical rebuilds with metric version tagging.

## 9. Reporting & Dashboards

### 9.1 Persona Dashboards

• District Leader: district snapshot, equity slices, school comparisons.   
• Principal: attendance & grade trends, teacher workload flags, intervention throughput.   
• Teacher: class mastery heatmap, missing/late triage, suggested interventions.   
• Specialist: caseload progress, session efficacy, scheduling conflicts.   
• Parent/Student: simplified progress view, goals, upcoming assessments; no internal notes.

### 9.2 Standard Reports

• Attendance Summary (daily/term), Chronic Absence Tracker, Grade Distribution by Category, Standards Mastery by Unit, Assessment Item Analysis, Intervention Outcomes, Communication Engagement Report, Pull‑Out Utilization.

### 9.3 Alerts & Subscriptions

• Role‑aware subscriptions (email/push/in‑app).   
• Alert digests to reduce noise; ability to snooze or adjust thresholds.   
• Every alert deep‑links to the underlying data and an action (message parent, create intervention, schedule meeting).

## 10. Workflow & Processes

### 10.1 Data Pipeline

• ELT cadence: streaming for class events, hourly micro‑batches for school aggregates, nightly rebuilds for district packs.   
• Orchestration with retries, dead‑letter queues, and alerting on failures.   
• Late‑arriving data handling with watermark logic and incremental backfill.

### 10.2 Validation & Certification

• Automated tests on metric definitions; sampling comparisons against SIS UI totals.   
• Data Steward sign‑off for high‑stakes dashboards (report cards, state submissions).   
• Change requests for metric formula updates with stakeholder approvals.

### 10.3 Intervention Workflow Integration

• From any alert/report, create an intervention plan; assign owner, dosage, and review date.   
• Auto‑generate follow‑up reminders via Phase D messaging.   
• On review, capture outcomes and close or adjust the plan; feed results to analytics for impact scoring.

### 10.4 Privacy, Access & Auditing

• Row‑level security ensures users see only permitted students/classes.   
• Sensitive fields (accommodations, discipline) masked by default; explicit share toggles.   
• Full audit (who viewed/exported what, when, and why) with retention ≥ 3 years.

## 11. Metrics & Acceptance

Acceptance Criteria:   
• Dashboards render within performance targets with district‑scale data.   
• Metrics match validated SIS counts within ±0.5% for like‑for‑like definitions.   
• Role‑based views restrict access appropriately in QA tests.   
• Alerts produce actionable next steps and deep‑links.   
  
Success Metrics:   
• ≥ 30% reduction in time spent preparing weekly admin reports.   
• ≥ 25% faster teacher identification of at‑risk students (measured by intervention start times).   
• ≥ 15% reduction in chronic absence within two terms for schools using alerts and interventions.   
• Parent portal engagement increases by ≥ 20% after digest adoption.

Forward Linkage: Phase F (Integrations & Platform) will expose analytics via standards (Ed‑Fi/OneRoster/LTI), SSO/IdP claims for role scoping, and secure data exchange for state reporting.

# Phase F – Future & Integrations (Business Requirements)

Version: 1.0 • Audience: Product, Engineering, Data, QA, Security, District Leadership

Dependencies: Phases A–E (Core SIS, Lessons, Assessments, Communication, Analytics). Phase F operationalizes interoperability, platform scale, and long‑term evolution.

## 6. Functional Requirements

### 6.1 Integration Domains & Use Cases

• Identity & Access: SSO (SAML 2.0, OIDC), provisioning (SCIM), role claims, class/roster claims.  
• Rostering & SIS Interop: OneRoster v1.1+ (CSV & REST), Ed‑Fi (v5.x) core, Clever/ClassLink launch and data sync.  
• LMS & Classroom Tools: LTI 1.3/Advantage deep links (assignments, grade return), Google Classroom/Microsoft Teams summaries.  
• State & District Reporting: standardized exports/APIs with schedule; schema mapping and validation; secure delivery (SFTP/API).  
• Assessment Vendors: secure import of benchmark/standardized scores; item‑level if provided; mapping to standards and students.  
• Payment/Fees: PCI‑aware integration for fees, field trips, activities; ledger export to district finance.  
• Messaging/Notifications: email (SMTP/API), SMS (opt‑in), push (APNs/FCM), language translation pipeline.  
• Analytics/BI: warehouse exports (row‑level secured), direct connectors (JDBC/ODBC), metric catalog API.

### 6.2 Platform Services

• Public REST/GraphQL APIs with versioning and role‑scoped tokens; SDKs (TypeScript, Python).  
• Eventing/Webhooks: subscription to domain events (student.enrolled, assignment.graded, attendance.marked, pullout.approved). Retries with backoff, signature verification, replay window.  
• Import/Export Pipelines: guided mappers, preview/validate, idempotent upserts, diff‑aware updates, error queues.  
• Plugin/Extension Model: safe UI extensions (embeddable panels) with permission scoping and audit trail.

### 6.3 Multi‑Tenant & Scale

• District‑aware tenancy with school isolation; per‑tenant encryption keys; configurable data residency.  
• Horizontal scale targets: 100k students/tenant, 1M+ events/day, 5k concurrent teacher sessions.  
• Throttling/rate‑limits per API key and per tenant; burst policies for nightly jobs.

### 6.4 Mobile & Offline

• Native apps (iOS/Android) for teachers/parents/students with push notifications.  
• Offline tolerant: attendance and grade entry queues with conflict resolution on reconnect.  
• Kiosk/Shared device mode for secure classroom check‑in (future).

### 6.5 AI‑Assisted Features (Human‑in‑the‑Loop)

• Scheduling Assistant: propose conflict‑free schedules and specialist pull‑out windows (teacher approval required).  
• Insight Assistant: summarize student 360, draft intervention plans, recommend catch‑up resources from Phase B library.  
• Authoring Assistant: lesson templates, assessment items, rubric suggestions with standards alignment hints.  
• Guardrails: explainability notes, confidence scores, never auto‑publish; always require human confirmation; audit prompts/outputs.

## 7. Non‑Functional Requirements

• Security & Compliance: FERPA/COPPA; SOC 2 Type II roadmap; data encryption in transit/at rest; secrets management; PII minimization in logs/exports; DLP scanning for outbound webhooks.  
• Privacy Controls: consent preferences (email/SMS/push); data retention policies; right‑to‑be‑forgotten workflows.  
• Reliability & Performance: API p95 < 300ms for standard reads; webhooks delivery ≥ 99% within 60s; import throughput ≥ 50k rows/min.  
• Observability: structured logs, distributed tracing, tenant‑level metrics, anomaly detection on syncs.  
• Backups & DR: point‑in‑time recovery; cross‑region replicas; RPO ≤ 15 min, RTO ≤ 2 hrs.

## 8. Data & Integration Model

### 8.1 Canonical Data Contracts

• Entities: District, School, Term, Calendar, User, Role, Class, Enrollment, Student, Teacher, Specialist, Assignment, Grade, Attendance, Assessment, Attempt, Standard, Lesson, Resource, Intervention, Message, Event.  
• IDs: stable UUIDs; external\_id mapping per connector; crosswalk tables for Ed‑Fi/OneRoster.  
• Timestamps: ISO‑8601 with timezone; server‑authoritative for assessments and attendance.

### 8.2 Standard Connectors

• OneRoster: courses, classes, enrollments; delta files; 1.1/1.2 REST with OAuth2.  
• Ed‑Fi: Student, Staff, Section, Enrollment, GradebookEntry, StudentSectionAttendance; change queries; vendor profile mapping.  
• LTI 1.3/Advantage: Deep Linking, Assignment & Grade Service; per‑platform registrations; kid‑safe scopes.  
• Clever/ClassLink: secure sync and SSO launches; app focus filtering.

### 8.3 Warehouse & Analytics Hooks

• Daily partitioned exports to district‑owned storage (Parquet/CSV) with dictionary and lineage metadata.  
• Role‑aware aggregation endpoints for dashboards; metric catalog API mirroring Phase E definitions.

## 9. API & Extensibility

• REST & GraphQL endpoints with explicit scopes (read:grades, write:attendance, admin:roster). API keys tied to tenant and app; per‑endpoint quotas; sandbox environment.  
• Webhooks: subscribe/unsubscribe APIs, HMAC signatures, replay within 7 days; dead‑letter inspection UI.  
• SDKs: TypeScript and Python clients; example apps; Postman/OpenAPI packages.  
• UI Extensions: embed surfaces (class, student, gradebook panels) with JWT‑based view tokens; content security policy.

## 10. Workflow & Processes

### 10.1 Onboarding & Migration

• Guided data import: map, validate, preview deltas, dry‑run, commit; undo window; reconciliation reports.  
• Coexistence: run parallel with legacy SIS/LMS; periodic sync; cutover checkpoints.  
• Change Management: stakeholder training, parent comms templates, go‑live readiness checklist.

### 10.2 Versioning & Backward Compatibility

• API versioning: semantic versions; deprecation policy with ≥ 12‑month overlap; migration guides.  
• Connector versioning: per‑integration adapters with mapping files; validation harness and certification tests.

### 10.3 Security Operations

• Vendor risk assessments; penetration testing cadence; coordinated disclosure policy.  
• Key rotation schedules; least‑privilege defaults; continuous permission review for service accounts.

## 11. Metrics & Acceptance

Acceptance Criteria:  
• Successful pilot with: OneRoster roster sync, LTI grade return, and SSO via OIDC for a multi‑school tenant.  
• API latency and webhook delivery meet NFR targets under district‑scale load.  
• Import pipeline completes 1M+ records with <0.5% errors, all with actionable diagnostics.  
• Data exports verified by district analytics team for schema and counts.  
  
Success Metrics:  
• 50% reduction in manual roster maintenance for pilot districts.  
• ≥ 90% of third‑party tools connected via standards (LTI/OneRoster/Ed‑Fi).  
• ≤ 1% monthly integration failure rate (auto‑recovered) with transparent incident reports.  
• Positive admin/teacher NPS (+30 or higher) regarding setup and reliability.

Dependency Linkage: Phase F assumes A–E are live; F exposes them via standards and APIs. Forward‑looking features (AI assistants, mobile offline) build on events and contracts defined here, ensuring no siloed functionality.

# SIS Add-On Package (Phases A–F): Cross-Phase Dependencies, Roles, Journeys, Schema, and Scale Notes

Version: 1.0 • Audience: Product, Engineering, Data, QA, Security, School/District Leadership

## 1) Cross‑Phase Dependencies Map

This section makes explicit how each phase depends on and feeds other phases. Use it to plan sequencing, API contracts, and QA scope.

A – Core SIS Foundation (District, Schools, Users/Roles, Classes, Scheduling, Attendance, Gradebook Baseline, Specialists)  
 ↳ Provides canonical IDs, rosters, calendars, attendance events, baseline gradebook entities.  
  
B – Curriculum & Lesson Planning  
 DEPENDS ON: A (classes, schedules, roles)  
 FEEDS: C (assessment blueprints/standards), D (weekly digests), E (coverage/pacing analytics)  
  
C – Advanced Grading, Assessment & Analytics Hooks  
 DEPENDS ON: A (gradebook containers, attendance), B (standards & lessons)  
 FEEDS: D (result notifications), E (item analysis, mastery), F (exports/APIs)  
  
D – Communication & Engagement  
 DEPENDS ON: A (rosters/roles), B (lesson summaries), C (grades/assessments)  
 FEEDS: E (engagement metrics), F (notification connectors)  
  
E – Reporting, Analytics & Insights  
 DEPENDS ON: A–D events and aggregates  
 FEEDS: A (intervention loops), D (nudges), F (warehouse/API contracts)  
  
F – Future & Integrations (SSO, OneRoster/Ed‑Fi, LTI, Data Warehouse)  
 DEPENDS ON: A–E stable contracts  
 FEEDS: External partners, district data platforms; informs versioning/back‑compat strategy.

Development Gate Checklist (per phase):

• API contracts documented (OpenAPI/GraphQL) and versioned.

• Event schema (for webhooks/CDC) published with examples.

• Test data sets spanning K–5 and 6–8 (incl. specialists, pull‑outs, accommodations).

• Role/permission mappings verified against the matrix in Section 2.

• Backward‑compatibility and migration notes recorded.

## 2) Unified Terminology & Role Permissions Matrix

Authoritative glossary and module‑by‑module permissions to eliminate ambiguity.

### Glossary (selected)

**District:** Top‑level tenant containing multiple schools and shared policies.

**School:** Organizational unit with its own calendar, bell schedules, and staff.

**Class (Section):** Teach‑by‑period entity: teacher + roster + schedule.

**Specialist:** Staff who deliver services via pull‑outs/push‑ins; may see multiple students simultaneously.

**Calendar Block:** Scheduled time window tied to a class or event.

**Lesson:** Planned instructional unit linked to standards and optionally to a calendar block.

**Assessment:** Measurable activity (quiz/test/performance) with scoring rules and optional online delivery.

**Assignment:** Gradebook item; may be created from a lesson or assessment.

**Intervention:** Action plan for a student with goals, dosage, owner, and review date.

**Parent‑friendly View:** Derived presentation excluding internal notes and accommodations.

### Role Permissions Matrix (summary)

Key = V(view), C(create), E(edit), A(approve), R(report/export). More granular scopes to be defined in IAM.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Module | Principal | Vice Principal | Dean | Staff Admin | Teacher | Specialist | Parent | Student |
| Roster & Classes | VCEA R | VCEA R | VCEA R | VCE R | V (own) | V (assigned) | V (child) | V (self) |
| Scheduling (Blocks) | VCEA R | VCEA R | VCEA R | VCE R | VCE (own) | VCE (own slots) | V (child) | V (self) |
| Specialist Pull‑outs | VCEA R | VCEA R | VCEA R | VCE R | A (approve/deny) | C E (request/edit) | V (child) | V (self) |
| Lessons | V R | V R | V R | V R | VCE (own/team) | V (student‑specific notes) | V (summary) | V (summary) |
| Assessments/Assignments | V R | V R | V R | V R | VCE (own) | V (assigned) | V (child) | V (self) |
| Gradebook | V R | V R | V R | V R | VCE (own) | V (assigned) | V (child) | V (self) |
| Attendance | VCE A R | VCE A R | VCE A R | VCE R | VCE (own) | V (sessions) | V (child) | V (self) |
| Messaging | VCE A R | VCE A R | VCE A R | VCE R | VCE (classes) | VCE (caseload) | VCE (child threads) | V (own) |
| Analytics | V R | V R | V R | V R | V (class) | V (caseload) | V (child summary) | V (self summary) |

Note: Staff Admin inherits Principal capabilities minus access to highly sensitive fields (discipline/IEP full text) unless explicitly granted.

## 3) End‑to‑End User Journeys (Representative)

**Admin – Start‑of‑Year Setup:** Create district and schools (A) → Import rosters via OneRoster (F) → Define bell schedules and calendars (A) → Publish curriculum maps and templates (B) → Configure category weights and grading periods (A/C) → Enable communication policies and digests (D) → Validate analytics dashboards (E).

**Teacher – Weekly Plan & Teach:** Open planner (B) → Drag lessons onto calendar (A) → Auto‑create linked assignments (A/C) → Specialist requests arrive; approve or propose alternative (A) → Teach; take attendance from block (A) → Post grades; auto‑notify parents (D) → View class mastery heatmap (E).

**Specialist – Manage Pull‑outs:** Propose sessions (A) → Teacher approves/denies with reason and alternate suggestion (A/D) → Conduct sessions; mark attendance (A) → Add notes tied to lesson/goal (B) → Review caseload impact report (E).

**Parent – Stay Informed:** Get weekly digest (D) → See upcoming lessons and assessments (B/C) → Receive alerts for missing work or grade drops (D/E) → Message teacher (D) → Track progress trends in portal (E).

**Student – Own Their Learning:** See schedule (A) → Review lesson summaries and resources (B) → Complete online assessments (C) → Check feedback and grades (C) → Receive nudges about missing work (D) → View progress dashboard (E).

## 4) Master Data Schema Overview (A–F)

High‑level ERD summary and table inventory across phases.

|  |  |
| --- | --- |
| Domain | Key Entities (IDs, Relationships) |
| Core SIS (A) | district(id)→schools(id); users(id, role); classes(id, school\_id, teacher\_id); enrollments(student\_id, class\_id); calendar\_blocks(id, class\_id, start, end); attendance(id, student\_id, date, status); gradebook\_categories(id, class\_id, weight); assignments(id, class\_id, category\_id, points, weight); grades(id, assignment\_id, student\_id, score, status) |
| Curriculum (B) | curriculum\_maps(id, school\_id, subject, grade\_level); unit\_plans(id, map\_id?); lesson\_plans(id, class\_id, date); lesson\_schedule\_links(lesson\_id, calendar\_block\_id); lesson\_assignment\_links(lesson\_id, assignment\_id); standards(id, code) |
| Assessments (C) | assessments(id, class\_id, blueprint, delivery\_mode); items(id, assessment\_id? bank\_ref); attempts(id, assessment\_id, student\_id); responses(id, attempt\_id, item\_ref, score); scoresheets(id, attempt\_id, total\_score); retake\_policies(assessment\_id, rule) |
| Communication (D) | threads(id); messages(id, thread\_id, sender\_id, recipients[]); deliveries(id, message\_id, channel, status, timestamps); digests(id, audience, schedule) |
| Analytics (E) | fact\_grades, fact\_assessments, fact\_attendance, fact\_schedule, fact\_pullouts, fact\_communications, fact\_interventions; dim\_student, dim\_teacher, dim\_class, dim\_school, dim\_subject, dim\_standard, dim\_calendar |
| Integrations (F) | external\_connections(id, type, creds\_ref); mappings(id, external, internal, transform); webhooks(id, topic, target\_url); exports(id, job, format, location); imports(id, source, status, diagnostics) |

ERD Notes: All domains use stable UUIDs; cross‑domain links via foreign keys; audit tables capture create/update with actor and reason; soft‑delete for recoverability where appropriate.

## 5) Scalability, Integration & Compliance Notes

• Partitioning & Indexing: time‑based partitions for facts; composite indexes (student\_id, class\_id, date) for hot paths.

• Caching: teacher dashboard and class roster caches with TTL; pre‑compute nightly aggregates for E dashboards.

• Event Bus: use domain events for D/E/F; ensure idempotency keys for imports/grade pushes.

• API Versioning: semantic versions with ≥12‑month deprecation windows; compatibility tests per connector.

• Observability: tenant‑scoped metrics; SLOs for API p95 and webhook delivery; anomaly alerts on sync drifts.

• Security: RBAC with row/column‑level policies; encrypted PII; consent preferences respected in all channels.

• Accessibility & i18n: WCAG 2.1 AA across portals; multilingual templates for D digests and messages.