INTERNSHIP PROJECT ASSIGNMENT: SCHOOL MANAGEMENT APPLICATION

Objective

This internship assignment is designed to develop students' ability to think and operate like professional software analysts and developers. Interns will be responsible for creating well-structured and comprehensive documentation for a School Management Application based on the provided Business Requirements Document (BRD) and Software Requirements Specification (SRS).

Task Overview

Interns must complete the following major components:

1. Business Requirement Document (BRD)
2. Software Requirements Specification (SRS)
3. Workflow Diagram
4. System Architecture Summary

All tasks must be created using the content already defined in the BRD–SRS.

Tasks & Deliverables

1. Create a Business Requirement Document (BRD)

Using the structured content provided, interns must prepare:

Purpose of the School App

The School Management Application is a comprehensive digital solution designed to efficiently digitize and streamline all core academic and administrative operations of a school. This includes student record management, attendance monitoring, examination processes, timetable scheduling, fee management, and structured communication among teachers, parents, and students. The primary objective of the system is to enhance operational accuracy, minimize manual intervention, and ensure greater transparency and efficiency across the entire school ecosystem.

Business Problems

Problem	Description	Impact
Manual attendance marking	Paper-based process	Errors and time=consuming
Poor Information tracking	Student data & marks stored manually	Data loss risk, inefficiency
Delayed fee updates	No transparent tracking for parents	Late payments & conflicts
Weak communication	Parents not informed on time	Lower student performance
Hard to manage results	Manual calculation & report cards	Mistakes & delays

Business Goals

Automate school operations

- Ensure timely fee collection, accurate tracking, and transparent payment records.
- Enable easy and effective monitoring of student academic performance.
- Reduce the overall workload of teachers and administrative staff through process automation.
- Enhance communication and collaboration between teachers and parents.
- Maintain centralized, secure, and reliable storage of all school data.

Stakeholder List (with responsibilities)

Stakeholder	**Responsibilities**
Admin	Full data control, user management
Principal	Monitor school performance and reports
Teachers	Attendance, exams, marks, assignments
Students	Learn, view marks & timetable
Parents	Track child progress & fee status
Accountant	Fee management & financial reports

High-Level Features of the System

Secure Login & Role-based Access

- Fees and Payment Management System
- Teacher Portal for Academic Activities and Class Management
- Parent Portal for Communication and Student Progress Monitoring
- Student Information Management System
- Timetable Creation and Scheduling
- Attendance Tracking and Management
- Examination Management and Result Generation

Deliverable: 2–4 page BRD PDF.

2. Create a Software Requirements Specification (SRS)

Interns must produce an SRS that includes:

A. User Roles & Descriptions

(Admin, Teacher, Student, Parent, Accountant, Principal)

User Role	**Permissions**
Admin	Full access to all modules
Principal	Reporting & performance approval
Teacher	Manage classes, attendance, marks
Student	View marks, timetable, notifications
Parent	View child's performance & fees
Accountant	Manage fee transactions

B. Functional Requirements

Modules:

Student Management

ID	**Function Requirement**
FR-01	Add / Edit / Delete student records
FR-02	Search students by class / section
FR-03	Upload student documents / photos

Attendance Management

ID	Function Requirement
FR-04	Teachers can mark attendance daily
FR-05	System auto-generates absence reports
FR-06	Parents notified if student absent

Timetable Management

ID	Function Requirement
FR-07	Admin / Teacher can create timetables
FR-08	Students and teachers can view timetable

Exam & Marks Management

ID	Function Requirement
FR-09	Teachers enter marks for every subject
FR-10	System generates grades automatically
FR-11	Students & parents view published results

Fees Management

ID	Function Requirement

FR-12	Manage fee structure, discounts
FR-13	Track payments (paid/pending)
FR-14	Auto reminders for fee due dates

Teacher Module

ID	**Function Requirements**
FR-15	View assigned subjects & classes
FR-16	Upload assignments / study materials

Parent Portal

ID	**Function Requirement**
FR-17	View child’s attendance, results
FR-18	View and pay fees

C. Non-Functional Requirements

Performance

Security

Scalability

Usability

Availability

Category	Requirement
Performance	Page load < 3 seconds
Security	Role-based access, encrypted records
Usability	Mobile-friendly responsive UI
Scalability	Support increasing users & data
Availability	System must be available 99% yearly

D. System Rules & Constraints

Only Admin can delete student records

- Fees cannot be deleted after receipt generation
- Attendance cannot be modified after approval

Deliverable: 4–8 page SRS PDF.

3. Create a Workflow Diagram

Students must produce a simple workflow diagram covering:

1. **Understanding Problem & Goals**
2. **Identify User Types**
3. **Gather Requirements**
4. **Create System Design**

This can be:

Drawn digitally, or

Hand-drawn and scanned

Start

↓

Identify Problems & Goals

↓

Identify User Roles (Admin, Teacher, Student, Parent…)

↓

Gather Requirements (BRD)

↓

System Requirement Specification (SRS)

↓

Design Architecture + Database + Wireframes

↓

Documentation and Review

↓

End

Deliverable: PNG or PDF workflow diagram.

4. System Architecture Summary

Interns must summarise the recommended architecture:

Frontend (React / Angular / Vue)

Backend (Node.js / Django / Spring Boot)

Database (PostgreSQL / MySQL / MongoDB)

Must include a 1-page explanation of how these components work together.

Layer	Technology	Purpose
Frontend	React / Angular / Vue	User Interface
Backend	Node.js / Django / Spring Boot	Business Logic + APIs
Database	MySql / PostgreSQL / MongoDB	Store school data

How it Works

User interacts through the frontend

- Frontend sends requests to backend APIs (login, attendance, fees…..)
- Backend retrieves or save data in the database
- Response returned and displayed to user

Benefits

Modern & scalable architecture

- Secure data access
- Easy future updates Cloud-Deployable solution

Deliverable: 1-page architecture summary.

Submission Format

All students must submit:

BRD (PDF)

SRS (PDF)

Workflow Diagram (PNG/PDF)

System Architecture Summary (PDF)

All files should be organized into a folder named:

StudentName_SchoolApp_Assignment

Evaluation Criteria

Criteria	Weight
Clarity of BRD	20%
Completeness of SRS	40%
Quality of Workflow Diagram	20%
Architecture Understanding	10%
Presentation & Formatting	10%

UI Wireframes (Login, Dashboard, Student Profile)

ER Diagram for Database

Risk & Mitigation Analysis