

EVS Student Management System

Presented By
Monisha Moudivender - 22 July 2025



Situation:



EVS College uses a Student Management System developed in-house in 2018. The system is web-based and is primarily accessed by administrative staff, faculty, and students. The system is web-based and accessible only through desktop browsers within the college premises. It is maintained by the college's internal IT team.

Features:

- **Student Registration and Profile Management:** The existing system allows basic student registration and profile updates require manual admin intervention
- **Course Enrollment:** Students can enroll in courses online
- **Attendance Tracking:** Faculty manually mark attendance, but there is no real-time dashboard for students or reports for parents.
- **Grade Management:** Grades are entered by faculty
- **Timetable and Exam Scheduling:** Timetables are uploaded manually and shared via PDF
- **Fee Management:** Fee records are manually updated by admins after offline payments
- **Notification System (email):** The system relies heavily on emails with no in-app or SMS alerts
- **Report Generation:** Reports are basic and static, generated manually by admin staff. There is no custom reporting, data export, or analytics capability for management decision-making.

The software helps manage academic records, and communication between students, faculty, and administration.



Problems

- **No Mobile Accessibility**
Students cannot access the system conveniently on mobile phones
- **Outdated User Interface**
The system interface not user-friendly, causing confusion especially for new students and faculty.
- **Manual Fee Collection Process**
Fee payment records are entered manually by the admin staff. This increases the chance of human error and delays.
- **Delayed Notifications**
Important updates like schedule changes or grade announcements are delayed due to reliance on email-only communication.
- **System Downtime During Peak Load**
The system slows down or crashes during admission or examination periods due to limited server capacity.
- **Lack of Integration with LMS Tools**
Faculty need to use separate platforms to share learning materials and conduct online assessments, resulting in duplicated effort.
- **Report Generation:** Reports are basic and static, generated manually by admin staff. There is no custom reporting, data export, or analytics capability for management decision-making.



Opportunities

- **Mobile Application Development**

Create Android and iOS apps access for students and faculty.

- **Modernize the User Interface**

Redesign the UI to improve navigation, reduce user errors, and enhance the overall user experience.

- **Integrate Online Payment Gateway**

Enable secure online payments and automatic fee reconciliation, reducing workload for the admin staff.

- **Implement Notification System**

Real-time alerts through SMS, mobile app, or system notifications

- **Cloud Migration**

Move the system to a cloud-based infrastructure for better scalability, data security, and system availability.

- **LMS Integration**

Connect with platforms like Moodle or Google Classroom to manage academic content and online assessments within a single platform.

- **Analytics and Reporting Dashboard**

Introduce dashboards that give management and faculty insights into student performance, attendance trends, and fee status.



Goals

- **Enhance Operational Efficiency**
Streamline administrative processes such as student registration, attendance, grading, and fee management.
- **Improve Student and Faculty Experience**
Provide a user-friendly, accessible platform to enhance engagement and satisfaction.
- **Enable Real-Time Access to Academic Data**
Ensure students, faculty, and administrators can access up-to-date information on academics and operations.
- **Support Decision-Making with Insights**
Offer automated reports and analytics to assist in academic and administrative planning.
- **Facilitate Seamless Communication**
Improve communication between students, faculty, and administration through automated alerts and notifications.



Project Objectives

- **Develop a Mobile-Responsive Platform or App**
Launch a mobile version of the system for both Android and iOS users.
- **Redesign User Interface**
Complete a UI/UX to make the system accessible.
- **Integrate Payment Gateway**
Automate fee collection and reconciliation by integrating an online payment system
- **Implement Notification System**
Enable SMS, email, and in-app push notifications for important updates.
- **Migrate to Cloud Infrastructure**
Shift to a scalable cloud platform to improve performance and reduce downtime by the next academic cycle.
- **Enable LMS Integration**
Allow faculty to link LMS tools (Google Classroom) with SMS by the end of the current semester.
- **Generate Custom Reports and Dashboards**
Build a dashboard module with downloadable academic and financial reports.



Success Criteria of the Student Management System

- **System Accessibility Across Devices**
The system is accessible via desktop and mobile (Android/iOS) platforms.
- **Improved User Satisfaction**
The new interface is user-friendly and intuitive.
- **Reduction in Manual Work**
Administrative tasks like fee collection, attendance, and grade entry are automated.
- **Real-Time Communication**
Push notifications, SMS, and emails are sent for critical updates.
- **System Performance and Uptime**
The platform remains stable during peak usage times like admissions or exam results.
- **Data Security and Compliance**
Role-based access and data encryption are implemented.
- **Custom Reporting Functionality**
The system can generate academic, attendance, and financial reports on demand.
- **Successful LMS Integration**
Faculty can access and upload course content from integrated LMS platforms.



Methods/Approach:

Business Analyst Contribution in the Student Management System Project

Requirement Elicitation & Analysis

- Identified primary and secondary stakeholders: students, faculty, administrative staff, IT support, and college management. Created a **stakeholder map** to define influence, interest, and communication frequency.
- Conducted interviews, prototyping, brainstorming, document analysis, observation, and surveys to understand their pain points and expectations from the new system.
- Collected and clarified functional and non-functional requirements for modules like registration, attendance, grade management, and notifications.

User Story Creation

- Translated business requirements into detailed user stories with acceptance criteria for each module (e.g., "As a student, I want to view my attendance history so that I can track my progress").
- Maintained and refined the product backlog in collaboration with the Product Owner and Scrum Master.
- Defined and reviewed acceptance criteria for each user story to guide developers and QA testers.



Process Mapping

- Created AS-IS and TO-BE process flows for key functions such as course enrollment, timetable scheduling, and fee payment.
- Used tools like Visio for UML diagrams.

Design Phase (Initial Sprints)

Support early design activities and sprint planning

- Collaborated with UI/UX designers to ensure screens meet user expectations.
- Validated wireframes and mockups with end users and incorporated feedback.
- Ensured that all user stories are aligned with business needs before sprint kickoff.
- Reviewed designs for modules such as student dashboard, grade entry, and course enrollment.



Sprint Execution (Development Phase)

- Acted as a bridge between technical teams and end users to ensure mutual understanding of requirements and progress.
- Facilitated daily stand-ups, sprint planning, backlog grooming, and sprint review meetings with clear documentation of feedback.
- Participated in **Sprint Planning Meetings** to:
 - Clarify user stories
 - Explain business value
 - Assist in effort estimation using story points
- Attended **Daily Stand-Ups** to track progress and answer clarifying questions from developers or testers.
- Presented developed features to stakeholders during **Sprint Reviews**.
- Collected **feedback and enhancement requests**.
- Logged changes as new user stories or updated backlog items.



UAT

- Designed **UAT test scenarios and scripts**.
- Coordinated testing sessions with actual users (faculty, students, admins).
- Logged feedback, tracked issues, and ensured fixes were implemented before go-live.

Risk & Dependency Management

- Helped identify risks like:
 - Data loss during migration
 - Resistance to new system adoption
 - Payment gateway delays
- Worked with Scrum Master to maintain a **Risk Register** and mitigation plan.
- Manage **dependencies** like cloud infrastructure readiness, third-party SMS APIs, and app store approvals.



Deployment & Training Phase

- Created **training guides** for end users.
- Conducted training sessions for:
 - Students (using the portal and app)
 - Faculty (uploading grades, scheduling)
 - Admins (fee management, reporting)

Post-Go-Live Support & Continuous Improvement

- Collected user feedback after launch and identified areas for improvement.
- Participated in backlog grooming to include new enhancement requests in future sprints.
- Helped measure **success criteria** (e.g., improved usage, reduced manual tasks, user satisfaction surveys).



Why Select Agile Methodology for the Student Management System Project

- **User-Centric Approach**

- Agile emphasizes **continuous feedback** from end users (students, faculty, admin staff), ensuring the system evolves based on real user needs.
- Frequent demos help stakeholders visualize progress and suggest changes early.

- **Iterative Development**

- Agile breaks the project into **small, manageable sprints**, allowing incremental development and delivery of core features like registration, fee management, and attendance tracking.
- This reduces risk and helps prioritize features based on immediate value.

- **Flexibility to Handle Changing Requirements**

- Educational institutions may request **changes mid-project** (e.g., mobile version, LMS integration).
- Agile handles this well through **backlog grooming** and sprint planning without derailing the whole project.

- **Faster Time-to-Value**

- With Agile, **basic modules can go live early**, allowing partial use (e.g., only student registration or attendance) while other features are still being developed.



- **Better Communication & Collaboration**
 - Agile encourages **regular communication** between developers, business analysts, testers, and users through daily stand-ups and sprint reviews.
 - This leads to better understanding and fewer misinterpretations of requirements.
- **Early Detection of Issues**
 - Continuous testing and reviews in each sprint enable the team to **identify bugs or design issues early**, reducing cost and effort of fixing them later.
- **Stakeholder Involvement & Transparency**
 - Frequent sprint reviews and retrospectives keep stakeholders informed, engaged, and **in control of the project progress**.



Resources Required for Student Management System Project

Human Resources (10 People)

Product Owner

Owns the product backlog, prioritizes features, gathers stakeholder input

Scrum Master

Facilitates Agile ceremonies, removes roadblocks, ensures Agile practices are followed

Business Analyst

Elicits requirements, writes user stories, manages backlog grooming

UI Designer, Developers, DevOps

Designs user-friendly interfaces, API, database, mobile app, coding, Manages pipelines, cloud deployment

QA Tester

Writes test cases, testing each sprint

DevOps Engineer

Manages pipelines, cloud deployment



Time Resources

Time allotted for this project is 8 months

1. Planning & Requirement Gathering

4 weeks

2. Design

4 weeks

3. Core Development (5–6 Sprints)

3 Months

4. Testing

2 Months

5. Deployment

4 weeks

6. Support & Maintenance

Ongoing



Technical Resources

Development and Test Servers

For coding and testing before deployment

Cloud Hosting

For live deployment and secure access

Backup and Recovery System

For regular backup of student records and system logs

Laptops/Desktops

For development, testing, and user access.

Servers or Cloud System

To host the Student Management System securely.

Database Software

To store student, fee, course, reports (e.g., MySQL, SQL Server).

Project Management Tools

Jira for sprint planning and tracking

Collaboration Tools

Slack, Microsoft Teams, or Zoom for daily stand-ups and reviews

Analytics Tools

Power BI, Tableau, or built-in dashboards for reporting



Financial Resources

Budget allotted for this project is 15000000

Software Development Cost

Developers, licensing tools, salary - 10000000

Hardware & Infrastructure

Servers, computers - 2500000

Training, test

Manuals, trainers, tools, database - 1000000

Maintenance & Upgrades

Bug fixes and improvements - 1500000



Risks in Student Management System Project

1. Requirement Risk
Incomplete, improper requirement, change in requirement
2. User Resistance
Faculty or admin may resist adopting a new system.
3. Data Security Risk
Unauthorized access to student records.
4. Integration Risk
Faculty or admin may resist adopting a new system.
5. Stakeholder management Risk
6. Budget Overrun
7. Testing Risk
Bugs may go undetected if testing is rushed
8. Data loss Risk
Loss of existing student data while migrating from old system.



Dependencies in Student Management System

1. Stakeholder Availability
2. IT Infrastructure
3. Third-Party Software
4. Data Availability
5. Role-Based Access Setup
6. Decision Approvals
7. Training Completion
8. Mobile App Store Approvals

THANK YOU

