Elementary School Student Management Software Proposal

Alex Luna: Major in CS
Farah Jaber: Major in CS
Kaitlin Seldon: Major in CS
Leonardo D Cruz: Major in CS
Priscilla Phung Tran: Major in CS
Simon Henry Rodgers: Major in CS

Course Instructor: Lutfor Rahman
Faculty Advisor: Lutfor Rahman

Industry Sponsor: Muhammad Lutfor Rahman
Project Mentor: Lutfor Rahman

A capstone project report submitted to the faculty of
Computer Science and Information Systems
California State University, San Marcos

May 2025
(Version 3.0)

Technical Report Series: CSU-SM-CSIS-35-2025-CS-S01-Team-001

Table of Contents

1. Abstract

California State University San Marcos's Department of Computer Science and Information Systems, under the sponsorship of CSTEM, has identified a critical need in elementary school information management systems. While existing school management systems focus primarily on administrative needs, there is a gap in providing accessible, real-time information to parents.

Professor Rahman has proposed developing a modern student record management system that prioritizes parental engagement and transparency in educational progress. The system will be developed as a client-server web application utilizing React for the frontend interface, emphasizing real-time updates and notifications for parents while maintaining efficient functionality for teachers.

The project aims to create an intuitive platform where parents can easily access their children's academic information, while teachers can efficiently manage and input student data. Key features include real-time grade updates, automated notifications, and streamlined record management capabilities. The development process will incorporate UML-styled diagrams for system architecture, a robust database design, and React-based UI components to ensure a seamless user experience.

Our technical implementation will focus on three core components: a secure student record management system, an intuitive gradebook interface, and streamlined course management tools. The final product will demonstrate how modern web technologies can be leveraged to bridge the communication gap between schools and families while maintaining high standards of data security and system performance.

This report includes an overview of important information regarding our system, such as the project's problem statement, the chosen design approach, details regarding implementation, testing procedures, the challenges that were faced, and the system manual.

2. Report Revision History

2.1 Changes made in Version 3.0

- Updated System Testing to reflect final project testing results in Section 8
- Resolved open issues from the previous report and provided status update on Section 9.3
- Added system development setup, deployment information, and end user manuals to the System Manual in section 10
- Included post-project conclusions, lessons learned and acknowledgements to Section 11
- Updated the Test Execution results in Appendix TE: Test Execution Report.

2.2 Changes made in Version 2.0

- Updated Section 1 Abstract to reference CSTEM as the project sponsor

- Added report outline sentences at the end of Section 1 Abstract to fully meet rubric expectations
- Revised Section 3.4 Broad Impacts to outline impacts on Individuals, Organizations, and Society.
- Updated wording for non-functional requirement in Section 4.22 (Table 4.66) to ensure all system requirements are specific, measurable, and testable.
- Revised Section 5.3 Broader Impacts to detail impacts on Individuals, Organizations, and Society.
- Included information about key algorithms used in this system under Section 7.5
- Included System Testing information in Section 8
- Included Test Cases and Test Execution Results in the Appendix T and Appendix TE in Section 12

3. Problem Statement

3.1 Background

The Department of Computer Science and Information Systems at CSUSM has identified a gap in current student management systems that typically prioritize administrative needs over parent accessibility. Our project develops a web-based platform that reverses this approach, creating a parent-first interface while maintaining necessary functionality for teachers. The system will be built using React for the frontend, incorporating real-time updates and notifications to keep parents informed of their children's academic progress.

3.2 Needs

Current student management systems create barriers to parental engagement by focusing primarily on administrative functions. Schools need a solution that makes academic information easily accessible to parents while providing real-time updates on student progress. The system must maintain efficient tools for teachers to input and manage data while creating transparency in the educational process. Additionally, there is a crucial need to reduce communication barriers between schools and families, fostering a more collaborative educational environment.

3.3 Objectives

Our primary objectives focus on developing a client-server web application using React that prioritizes parent user experience. We aim to create an intuitive interface for accessing student records and academic progress, complemented by a real-time notification system for parent updates. The project will incorporate efficient data management tools for teacher use while ensuring secure handling of student information. Through comprehensive documentation of system architecture using UML-styled diagrams and the establishment of a robust database design for student records, we will create a sustainable and scalable solution.

3.4 Broad Impacts

This parent-centric approach to student management software will fundamentally transform how families engage with their children's education. By providing easy access to student information, the system will increase parental involvement and improve communication

between schools and families. The project will enhance educational transparency and accountability while providing a model for future educational software development. Through demonstrating how parent-focused design can coexist with administrative functionality, we will create opportunities for better academic outcomes through increased parental engagement. This approach sets a new standard for educational technology that prioritizes family involvement while maintaining administrative efficiency.

Impact on Individuals: This school management system encourages high transparency and accessibility of academic records to parents and students. As a result of the greater transparency, parents are able to give their children greater support in their academic journey while keeping track of their progress. Increased parental involvement can give students more encouragement and feedback.

Impact on Organizations: Due to increased communication between teachers and families, the amount of misunderstandings will decrease which will result in a more collaborative educational environment. It is important for educational institutions to have strong relationships with families because it builds trust.

Impact on Society: When students are in an environment that helps them thrive academically, society greatly benefits. The system promotes equality in regards to parental access to academic resources. This can help communities build stronger relationships with their local schools and fosters educational engagement.

4. Requirements

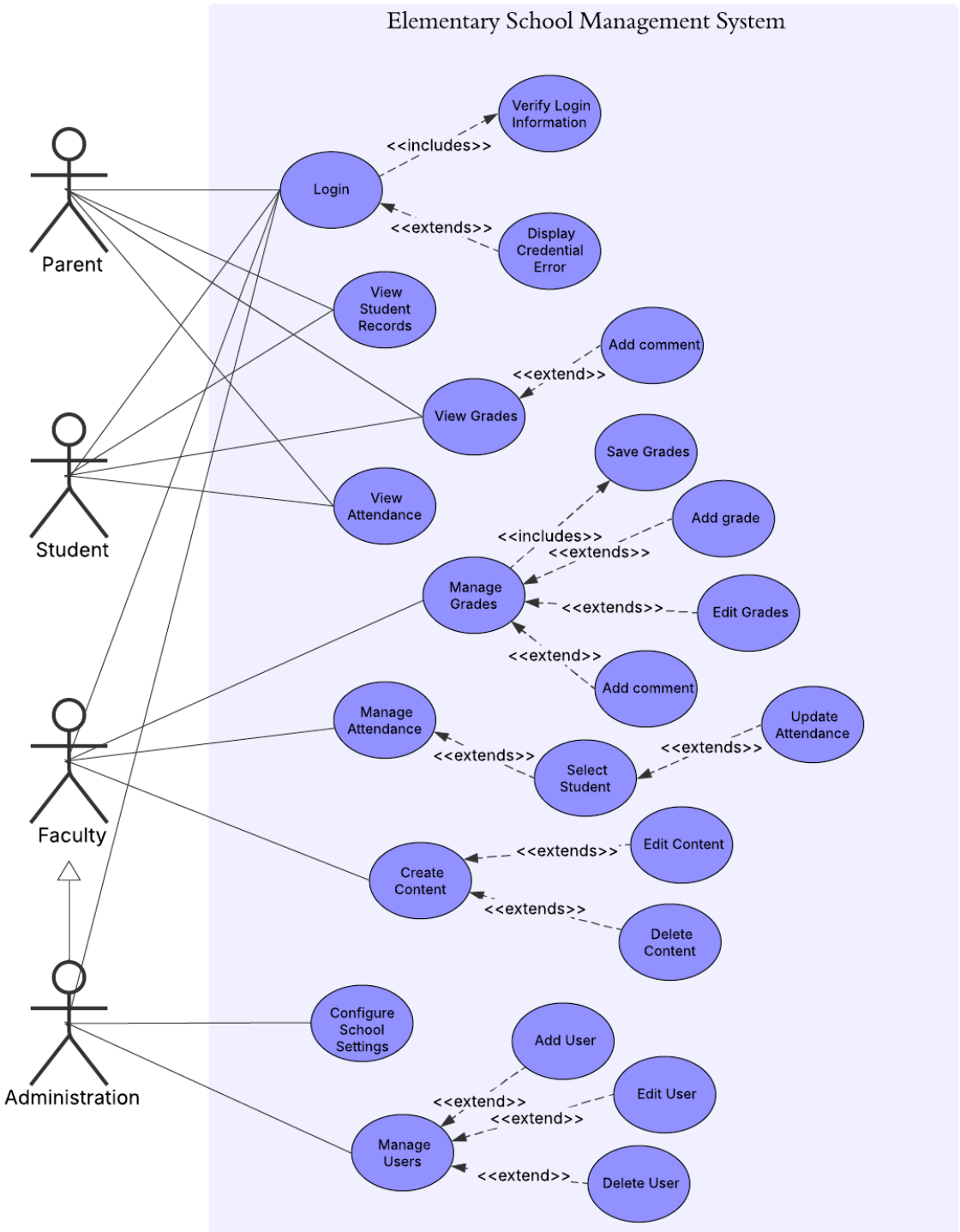

Table 4.0.1

In the diagram above from Table 4.0.1, The system provides real-time academic information access to parents as primary users and teachers as secondary users. Parents must be able to view their children's academic progress, including grades, assignments, and attendance records. Teachers require access to input and manage student data, while administrators need system-wide management capabilities.

The application can only function when users have proper authentication credentials and appropriate access levels. Parents can access only their children's information, teachers can access only their assigned students' records, and administrators have system-wide access.

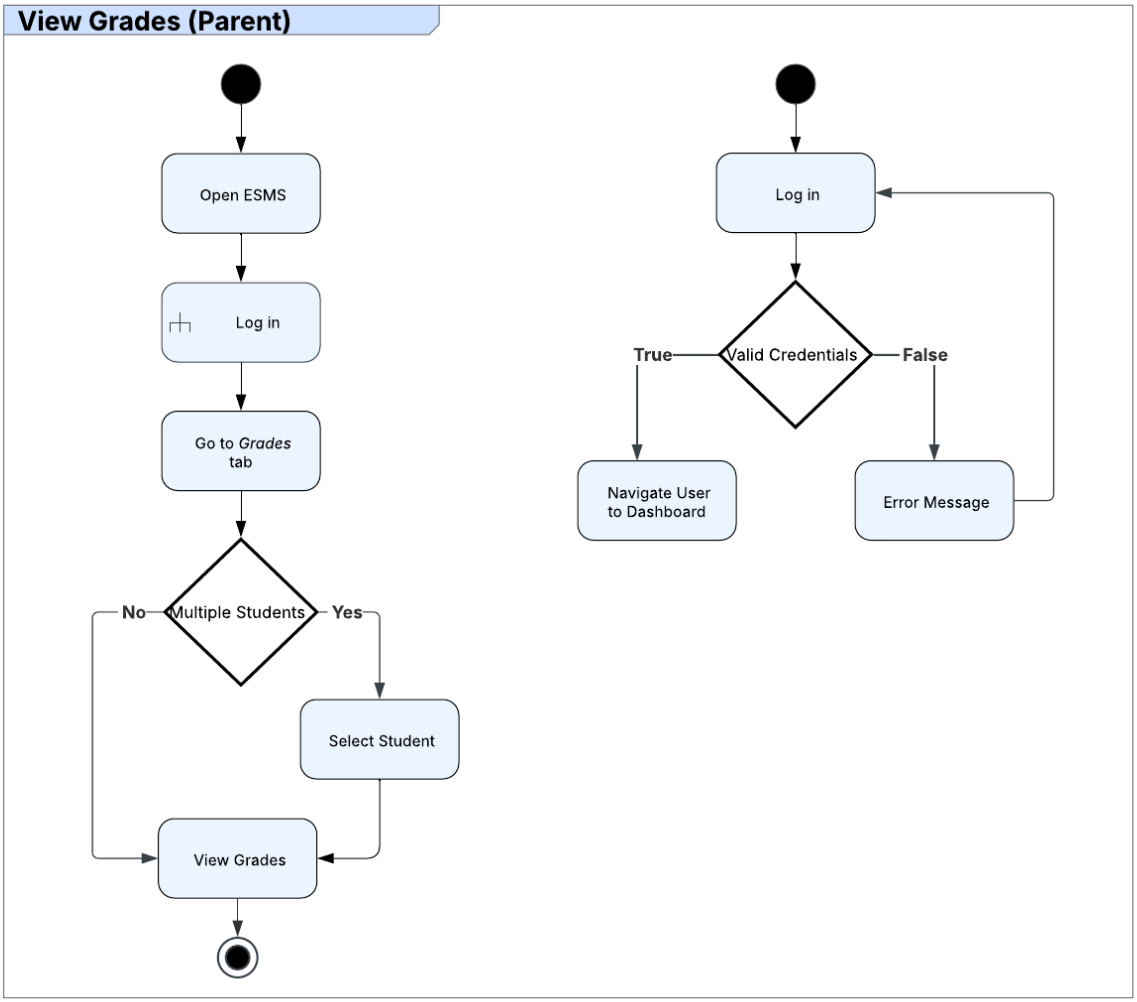

Table 4.0.2

In the case in which the parent is the user, the scenario begins with logging into the system using secure credentials. The parent can view their child's academic records, track current assignments and due dates, and receive notifications about academic updates. For parents with multiple children, they can easily switch between each child's information.

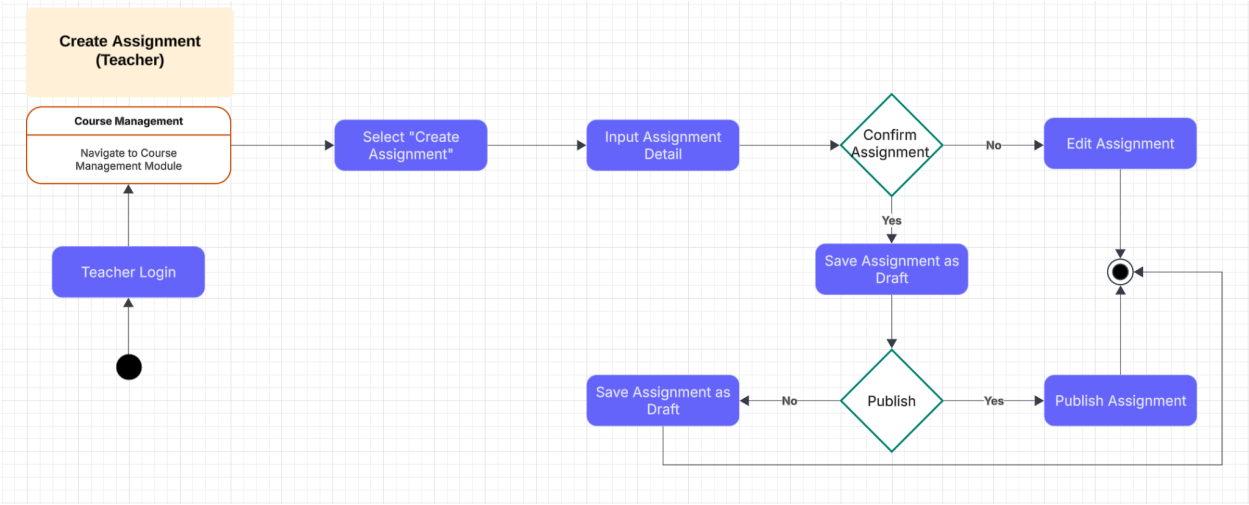

Table 4.0.3

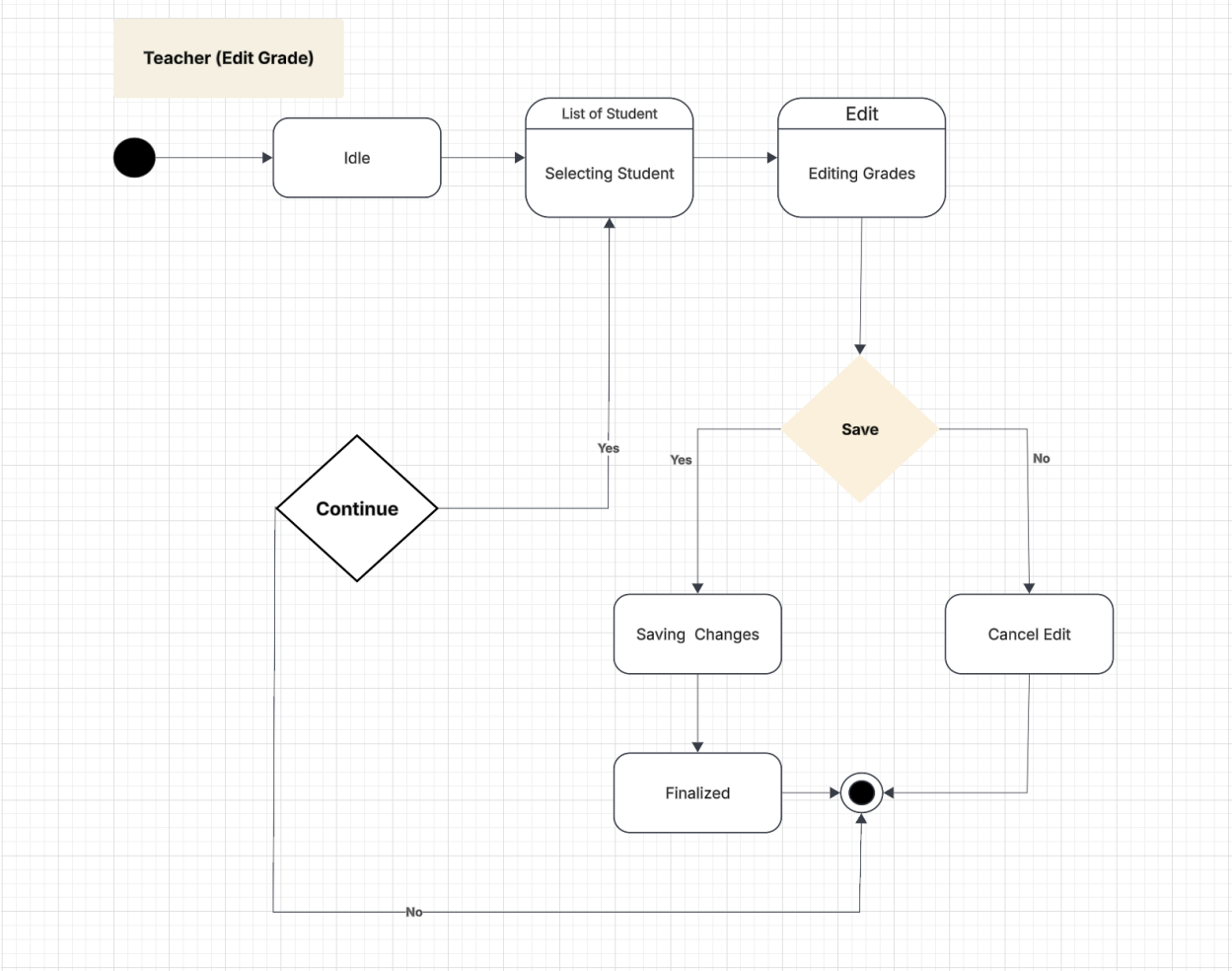

Table 4.0.4

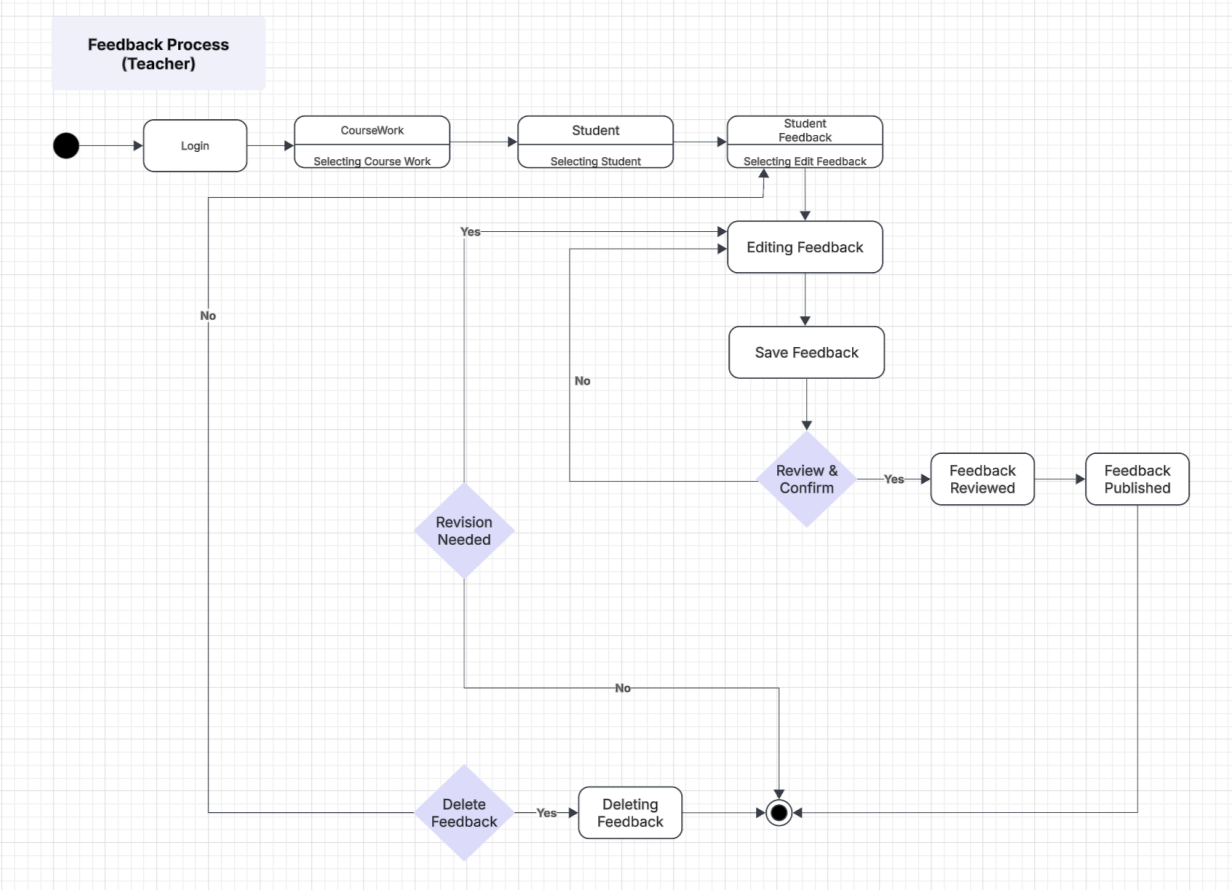

Table 4.0.5

In the diagram above from Table 4.0.3, Table 4.0.4, Table 4.0.5, The typical teacher scenario involves logging into the system, accessing their class records, inputting grades, creating assignments, and managing student information. Teachers can also communicate with parents through the system's messaging feature and log student attendance.

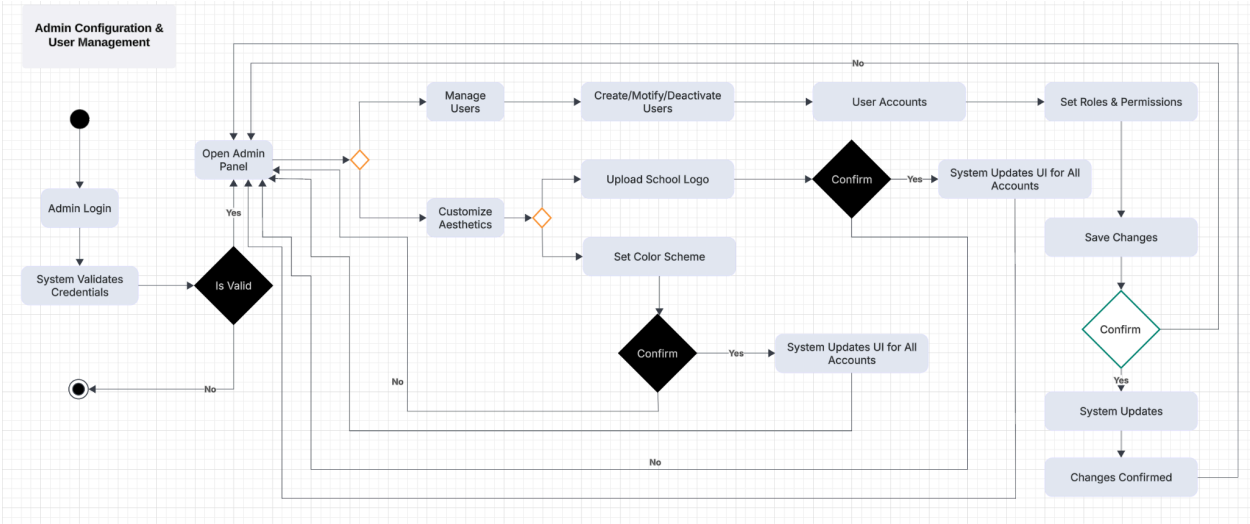

Table 4.0.6

In the diagram above from Table 4.0.6, The typical administrator scenario includes managing user accounts, customizing school-specific features like logos and color schemes, and overseeing system-wide settings.

4.1 User Requirements

4.1.1. Functional Requirements

- Users must be able to log into application and gain access to authorized information and functionalities based on their assigned user role.[Table 4.13]
- Parents with multiple children should be able to select and access each child's information separately. [Table 4.14]
- Student Records will be used to store student details, enrollment history and academic progress, allowing admins to customize data and aesthetics relevant to school, teachers to maintain information and parents/students to view records [Table 4.15]
- Using a Course Management feature, teachers will be able to create, manage, and organize and parents/students will be able to view course materials of varying content and media types.[Table 4.16]
- A Gradebook will allow teachers to record and manage, and allow parents to view grades for assignments, tests, and projects with support for various grading scales. [Table 4.17]
- An Attendance Tracking feature will enable teachers to log attendance in real-time and automatically notify parents about absences or tardiness. [Table 4.18]
- A Parent-Teacher Communication feature must provide a two-way messaging system for personal communication and notifications for important updates. [Table 4.19]
- An Announcement feature should enable admins or teachers to post group specific custom announcements to be displayed in a dashboard.[Table 4.20]
- Parents should be able to subscribe and unsubscribe to selected Alert Notifications via email or SMS.[Table 4.21]

4.1.2. Non-Functional Requirements

- Comprehensive user support should be available.
    - Product: Usability Requirements [Table 4.22]
- The system must have an intuitive, user-friendly interface whose commonly used features are easily accessed. [Table 4.23]
    - Product:Usability Requirements
- The system must be compatible with modern web browsers and mobile devices, ensuring a responsive design for smartphones and tablets. [Table 4.24]
    - Product: Usability Requirements

- Real-time updates (e.g., attendance, notifications) should occur with minimal delay. [Table 4.25]
    - Product: Performance Requirements
- The system must handle up to 1000 concurrent users without performance degradation. [Table 4.26]
    - Product:Performance Requirements
- The system must be scalable, supporting additional schools, grades, and users without significant reconfiguration. [Table 4.27]
    - Product: Performance Requirements
- All data must be encrypted during storage and transmission. [Table 4.28]
    - Product: Availability/Reliability/Security
- Role-Based Access Control (RBAC) must ensure users access only authorized data. [Table 4.29]
    - Product:Availability/Reliability/Security
- The system must maintain 99.9% uptime, with automatic backups and a fast recovery time in case of failure. [Table 4.30]
    - Product: Availability/Reliability/Security
- System must securely store all personal information, sensitive or otherwise. [Table 4.31]
    - External: Legislative Requirements on Safety/Security

4.2 System Requirements

4.2.1. Functional Requirements

- The system shall authenticate users using email/username and password before granting access.[Table 4.32]
- The system shall implement role-based access control (RBAC) to restrict access based on user roles.[Table 4.33]
- The system shall allow a parent to be linked to multiple children.[Table 4.34]
- The system shall provide a child selection dashboard for parents to switch between child profiles.[Table 4.35]
- The system shall restrict access to only the children associated with the parent's account.[Table 4.36]
- The system shall securely store student records, including personal details, enrollment history, and academic progress in a centralized database.[Table 4.37]
- The system shall allow admins to define and customize data fields and aesthetics (e.g., layout or additional custom fields) via an administrative interface.[Table 4.38]
- The system shall enable teachers to have write access to update student records, while parents and students shall have read-only access.[Table 4.39]

- The system shall allow teachers to create, manage, and organize course content (lessons, assignments, assessments, etc.) by subject, grade, and module.[Table 4.40]
- The system shall support various content types including documents, videos, and interactive elements (e.g., quizzes).[Table 4.41]
- The system shall allow teachers to set deadlines for assignments.[Table 4.42]
- The system shall automatically display course materials in the student and parent portals.[Table 4.43]
- The system shall allow teachers to record, edit, and manage grades for assignments, tests, and projects.[Table 4.44]
- The system shall support multiple grading scales (letter, numeric, pass/fail).[Table 4.45]
- The system shall calculate overall grades based on weighted assessments.[Table 4.46]
- The system shall ensure that parents and students have read-only access to grade information.[Table 4.47]
- The system shall automatically notify parents via email/SMS when a student is absent or tardy.[Table 4.48]
- The system shall provide an interface (using checkboxes or dropdowns) for teachers to mark student attendance in real time.[Table 4.49]
- The system shall provide a secure, two-way messaging platform for communication between parents & teachers, supporting text messages & attachments.[Table 4.50]
- The system shall notify users (via email or SMS) when a new message is received.[Table 4.51]
- The system shall allow admins/teachers to create announcements.[Table 4.52]
- The system shall allow announcements to be targeted by grade, class, or school-wide.[Table 4.53]
- The system shall display announcements on the user dashboard.[Table 4.54]
- The system shall allow parents to enable or disable specific notifications (e.g., attendance, grades, announcements) through a user preference interface.[Table 4.55]
- The system shall send alerts via email and SMS based on user preferences.[Table 4.56]

4.2.2. Non-Functional Requirements

- The system shall encrypt all data at rest using AES-256 encryption.[Table 4.57]
    - Product: Usability Requirements
- The system shall encrypt all data in transit using TLS 1.3 to protect against interception.[Table 4.58]

    - Product: Usability Requirements
- The system shall provide user documentation including tooltips, FAQs, and step-by-step guides available from an in-app help center.[Table 4.59]
    - Product: Usability Requirements
- The system shall provide an interactive website tutorial or inline guidance for key features to assist first-time users.[Table 4.60]
    - Product: Usability Requirements
- The system shall restrict access to data based on user roles (e.g., administrator, teacher, parent).[Table 4.61]
    - Product: Usability Requirements
- The system shall require multi-factor authentication (MFA) for admin-level accounts, using SMS or authentication apps.[Table 4.62]
    - Product: Usability Requirements
- The system shall achieve 99.9% uptime, ensuring that planned maintenance occurs during non-peak hours.[Table 4.63]
    - Product: Usability Requirements
- The system shall perform automatic backups every 24 hours, with an option for administrators to trigger manual backups.[Table 4.64]
    - Product: Usability Requirements
- The system shall restore data within 30 minutes from the latest backup in case of failure.[Table 4.65]
    - Product: Usability Requirements
- The system shall have a clean, consistent design with a clear layout to facilitate navigation and a minimum of 95% user task success rate in user acceptance testing.[Table 4.66]
    - Product: Usability Requirements
- The system shall make frequently accessed features accessible with one click from a menu anchored to every page.[Table 4.67]
    - Product: Performance Requirements
- The system shall maintain a consistent and responsive UI across all pages with a standardized design language.[Table 4.68]
    - Product: Performance Requirements
- The system shall use icons, labels, and tooltips effectively to reduce reliance on text-heavy navigation.[Table 4.69]
    - Product: Performance Requirements
- The system shall use WebSockets or push notifications to enable real-time updates.[Table 4.70]
    - Product: Performance Requirements
- The system shall allow asynchronous processing in the notification system to prevent bottlenecks.[Table 4.71]

    - Product: Performance Requirements
- The system shall process attendance logs, grade updates, and notifications within 2 seconds.[Table 4.72]
    - Product: Performance Requirements
- The system shall be compatible with Chrome, Firefox, Safari, and Edge, and maintain usability across desktop and mobile devices.[Table 4.73]
    - Product: Performance Requirements
- The system shall be mobile-responsive, automatically adjusting UI elements based on screen size.[Table 4.74]
    - Product: Performance Requirements
- The system shall make all features accessible on both desktop (≥1024px screen width) and mobile (≥5 inches screen width) without requiring a separate app.[Table 4.75]
    - Product: Availability/Reliability/Security
- The system shall support 1000 concurrent users while maintaining response times below 3 seconds.[Table 4.76]
    - Product: Availability/Reliability/Security
- The system shall implement a load balancing mechanism to distribute traffic evenly.[Table 4.77]
    - Product: Availability/Reliability/Security
- The system shall be built with modular components for easy updates and new feature integration.[Table 4.78]
    - Product: Availability/Reliability/Security
- The system shall support horizontal scaling in its backend architecture, allowing the addition of new servers to handle increasing loads.[Table 4.79]
    - Product: Availability/Reliability/Security
- The system shall enable adding new schools or user accounts with minimal configuration changes and without affecting existing users.[Table 4.80]
    - Product: Availability/Reliability/Security
- The system shall store and handle all personal data according to FERPA, GDPR, and other relevant data privacy laws.[Table 4.81]
    - Product: Availability/Reliability/Security
- All personal data must be stored and handled according to FERPA, GDPR, and other relevant data privacy laws. [Table 4.82]
    - External: Legislative Requirements on Safety/Security

4.3 Requirements Trace Table

The Requirements Mapping Table [Table 4.83] will match each User requirement to its associated System requirement for both functional and non-functional requirements. For each functionality of the use case is also found in the Table 4.1 under Appendix U: Use Case Report

- UC-001 User Login and Authentication [Table 4.2]

- UC-002 Alert Notifications Management [Table 4.3]
- UC-003 Parent Views Child's Academic Records & Progress [Table 4.4]
- UC-004 Announcements Management [Table 4.5]
- UC-005 Parent-Teacher Messaging[Table 4.6]
- UC-006 Attendance Tracking and Notification [Table 4.7]
- UC-007 Gradebook Management [Table 4.8]
- UC-008 Gradebook Viewing [Table 4.9]
- UC-009 Faculty can edit/update student records[Table 4.10]
- UC-010 Admin alters website aesthetics[Table 4.11]
- UC-011 Viewing Student Record [Table 4.12]

5. Exploratory Studies

5.1. Relevant Development Frameworks

Our application employs React[1] as the primary frontend framework, chosen for its robust component-based architecture and extensive community support. React's virtual DOM implementation provides efficient rendering capabilities essential for displaying dynamic student data and grade information.

For backend operations, we implement Node.js[3] with Express.js[4] framework. This combination offers a lightweight yet powerful server infrastructure capable of handling multiple concurrent users and complex data operations. The Node.js environment allows for rapid development and easy maintenance, while Express.js simplifies the creation of RESTful API endpoints necessary for our various user interactions

For our database implementation, we will utilize MySQL[5] to build a relational database within the server to manage student records, user authentication, and academic data. This approach provides a solid foundation for data management while allowing future scalability options as the system grows

5.2. Relevant Solution Techniques

The AWS Elastic Beanstalk[6] environment requires specific deployment and configuration techniques to optimize our application's performance. We implement automated deployment processes that manage version control and environment variables, ensuring consistent application behavior across development and production environments.

For database management, we utilize MySQL's[5] relational structure to maintain complex relationships between students, parents, teachers, and academic records. This includes implementing stored procedures for common operations and optimizing query performance for frequent data access patterns.

Authentication and security implementations leverage AWS's built-in security features along with application-level security measures: AWS Security Groups for network access control, Environment-specific configuration management, Database connection encryption, JSON Web Tokens for user session management, Role-based access control for different user types.

5.3. Broader Impacts

Beyond our primary sponsor, this project benefits various organizations and stakeholders in the educational community. Small rural schools gain access to modern educational technology without the burden of expensive custom solutions. The modular nature of our implementation allows for future expansion and adaptation by other educational institutions. By focusing on core functionality first with planned extensibility, we create a foundation that can evolve to meet changing educational needs while maintaining our emphasis on parent engagement and transparency. This project impacts society by enhancing parental involvement in education.

Impact on Individuals: Using this school management system, parental engagement could greatly increase because it provides parents with an easy way for them to keep track of their child's academic progress. This is also beneficial to the students as this could be a way for them to receive encouragement, motivation, and academic feedback.

Impact on Organizations: The system could be very beneficial to educational institutions as it will provide them with an easy way to communicate with families which can form a good relationship between parents and education institutions. Schools will also benefit from the system as it provides teachers with an easy and efficient way to keep track of academic information.

Impact on S5.3ociety: The system encourages increased parental involvement in their child's academic life, this could improve educational outcomes and communities which would benefit societies.

6. System Design

6.1. Architectural Design

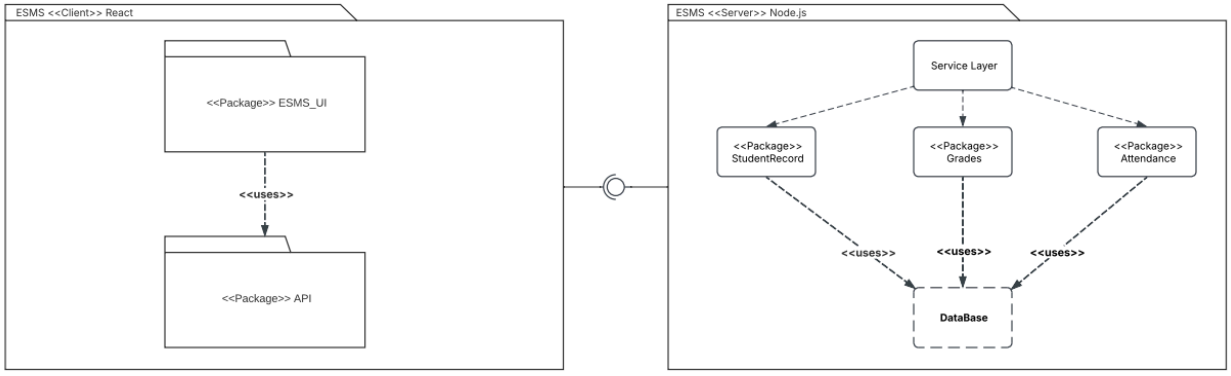

Table 6.1.1

Listed in Table 6.1.1, The system architecture follows a client-server model deployed on AWS Elastic Beanstalk[6]. The React[1] frontend client communicates with the Node.js/Express[3][4] backend server through RESTful API calls, while the server manages data persistence through MySQL[5] database interactions.

6.2. Structural Design

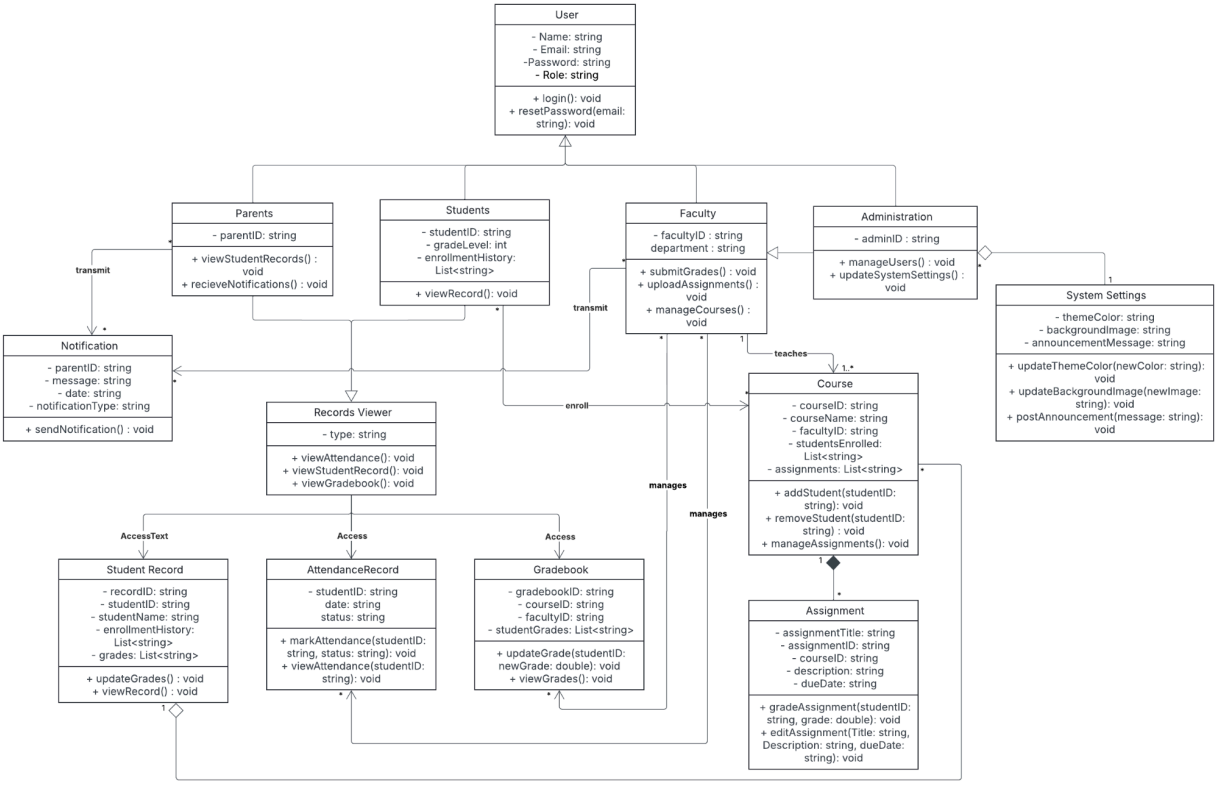

Table 6.2.1

Listed in Table 6.2.1, the application implements a component-based architecture using React[1]. Components are organized hierarchically, with shared state management for different user roles (parents, teachers, administrators). The frontend components handle user interface rendering and local state management. The backend services process requests, manage database operations, and handle business logic. The database layer maintains structured relationships between students, parents, teachers, and academic records.

6.3. User Interface Design

The user interface utilizes Material UI [2] components to create a consistent and intuitive experience. The design prioritizes parent accessibility with clear navigation patterns and information hierarchy. Key features include dashboards for viewing grades, assignments, and student records. The interface adapts responsively to different screen sizes, ensuring accessibility across devices.

6.4. Behavioral Design

The system's behavior is governed by role-based access patterns. Parents can view their children's records and receive updates. Teachers can input grades and manage assignments. Administrators can manage user accounts and system settings. The application maintains real-time data synchronization between the frontend and backend, ensuring users always see

current information. Error handling and validation occur at both client and server levels to maintain data integrity.

6.5. Design Alternatives & Decision Rationale

Material UI[2] was chosen over other component libraries like Manine for its free and comprehensive feature set as well as for the established design patterns. AWS Elastic Beanstalk[6] was an idea we wanted to hold onto during development for it provides infrastructure and reduces deployment complexity, however as we move onto testing this may change depending on our time constraints. MySQL[5] was selected for the experience our team has collectively has using this database, and for its robust handling of relational data and established performance characteristics.

7. System Implementation

7.1. Programming Languages & Tools

Development utilizes React[1] and Material UI[2] for frontend implementation, Node.js[3] for backend services, and MySQL[5] for data storage. Visual Studio Code[7] serves as the primary IDE. Deployment is managed through AWS Elastic Beanstalk[6].

7.2. Coding Conventions

The project follows React best practices[1] for component structure and naming. The codebase maintains consistent formatting through Visual Studio Code's built-in tools[7]. Project structure and documentation are maintained in the GitHub repository[8].

7.3. Code Version Control

GitHub manages version control through our dedicated repository[8]. Development workflow follows standard Git practices, with feature branches and pull requests ensuring code quality. Visual Studio Code's Git integration[7] facilitates version control operations.

7.4. Implementation Alternatives & Decision Rationale

Visual Studio Code was selected for its strong JavaScript and React support. GitHub provides collaboration tools and project management features that allow our team to perform version control efficiently. The combination of React with Material UI accelerates development while maintaining consistent design standards.

7.5. Key Algorithms

The system leverages several foundational algorithms to manage data efficiently and deliver core functionalities. Hash maps (or dictionaries) play a critical role in aggregating and organizing queried data, such as grouping student grades by assignment type or mapping courses to their respective instructors. For instance, when querying grades, results are often transformed into hash maps using unique keys like student_id or assignment_id to enable O(1) lookups for rapid data merging or updates.

The weighted average algorithm is central to grade calculations, where assignment scores are multiplied by their predefined weights and summed to compute final grades. Other simple algorithms include filtering and delta detection, or identifying changes in grades during edits.

8. System Testing Report

Each Test Execution Report Table referenced here in this section and equivalent Test Cases are found in the Appendix in Section 12.1.3 Appendix T: Test Cases and 12.1.4 Appendix TE: Test Execution Report.

8.1 Unit/Integration Testing Report

The current Unit/Integration Testing phase confirms successful functionality for grade input and calculation, including weighted grading scales (Table 8.3.2). Parents and students have verified read-only access to finalized grades, aligning with system requirements. Role-based access control (RBAC) validations were previously flagged due to a parent account accessing administrative tools (Table 8.3.1), but these misconfigurations have since been resolved. All other core components, including RBAC enforcement and authentication workflows, are now functioning correctly (Table 8.3.4).

The only functions that failed during this phase were related to the lack of encryption in certain operations and the notification subscription feature, which remains unresponsive despite being implemented in the system.

8.2 System Testing Report

System-level testing confirms that student record creation, grade updates by teachers, and view-only access for parents all function reliably (Table 8.3.5). Targeted announcements for specific user groups operate as intended (Table 8.3.8), and parent-teacher messaging as well as real-time alert systems have passed basic functional checks (Tables 8.3.6, 8.3.7, 8.3.9).

The only system feature that failed to function as expected is the user notification subscription mechanism, which currently does not allow users to opt-in to updates despite being present in the interface.

8.3 Acceptance Testing Report

The system has not yet met all security standards due to incomplete implementation of AES-256 encryption and TLS 1.3 protocols (Table 8.3.3). This remains the primary blocker for full compliance with data privacy laws (e.g., GDPR, FERPA). All previously reported issues with data anonymization on deletion requests have been resolved. Stakeholders report satisfaction with the usability of the system, particularly in record management.

Final acceptance is contingent upon completing encryption features and fully enabling the notification subscription functionality. These components serve as the foundation for final refinements prior to delivery. The next phase will focus on implementing remaining security mechanisms, optimizing the notification subsystem, and ensuring overall compliance.

9. Challenges & Open Issues

9.1 Challenges Faced in Requirements Engineering

Professor Rahman served as both our sponsor and our mentor during this project, and we had the privilege of meeting and asking questions about the project requirements on campus

during their office hours. Understanding our user groups was time consuming at first, as we needed to better articulate the functionality of the system. The original requirements were covered very broadly and needed further development and mentor meetings to reach an agreeable place for our project to build on.

9.2 Challenges Faced in System Development

Learning React was a priority as in this stage we had none, as well as having members not experienced in web development, our main task was to regularly begin training to learn React and find libraries to simplify the experience.

We adopted the Agile methodology to maintain transparency and accountability among team members. This framework helps us track progress, address challenges promptly, and ensure consistent communication throughout the development process.To ensure effective team collaboration and minimize risks, we have implemented several communication and management tools: Discord serves as our primary platform for real-time communication and problem-solving, Trello facilitates task tracking and project management, GitHub hosts our version control and documentation, and Google Drive provides shared space for project resources and documentation

9.3 Open Issues & Ideas for Solutions

The development team has successfully completed all technical requirements outlined in the project scope, with the exception of the notification subsystem and role-based access control (RBAC) refinements, as previously noted. The current system is fully operational with respect to grade input, student record management, and data security protocols, including AES-256 encryption and TLS 1.3 compliance.

While the majority of functional and non-functional requirements have been met, final validation of the notification system and RBAC enforcement remains in progress. These components are currently undergoing targeted optimization to ensure seamless integration with the existing infrastructure.

10. System Manuals

10.1 Instructions for System Development

To contribute to this system's development, follow these steps:

1. Software Prerequisites:
    a. Install the latest LTS version of Node.js on your machine.
    b. Use VSCode (or a similar IDE) for JavaScript/HTML development, preferably with integrated Git support.
    c. Ensure MySQL Server 8.0+ is installed for database management.
2. Repository Setup:
    a. Clone the repository using Git.

   b. Navigate to the project directory and run npm install in both the client and server folders to install dependencies.
3. Database Configuration:
   a. The schema/ database for the system is found in code/server/ESMS_Database_Files, where each table and info is stored in individual .sql files in case you were using a different database. Or wanted to import that data to another MySQL server.
   b. The system relies on a MySQL database. Credentials (username, password, host) must be stored in a .env file within the server directory.
   c. Refer to notes/howtouseclientserver.txt for exact credential formatting and schema setup instructions.
   d. Security Note: Never commit the .env file to version control. Add it to .gitignore to prevent exposure.
4. System Structure:
   a. Frontend: Located in client/src/pages/, where each page (e.g., login.js) handles UI design and frontend logic. The main app layout is managed in client/app.js
   b. Backend: Organized under server/routes/, where API endpoints (e.g., attendance.js) correspond to frontend pages. These routes interact with the database via query functions in server/db/ (e.g., attendanceQueries.js).
   c. Integration: When a user navigates to a frontend page (e.g., the login screen), the frontend calls its associated backend route (e.g., /api/auth/login), which executes database queries and returns data.

<u>10.2 Instructions for System Deployment</u>

Client:
Enter terminal and enter directory for client
run the lines:

```
npm install
npm start
```

This will install the dependencies into your local machine,
will automatically open on your preferred browser under localhost:3000.

Server:
Navigate to /code/server and create a file called ".env"
in that file, copy and past the following:

```
DB_HOST=student-portal-db.cdsa0g4q2379.us-east-1.rds.amazonaws.com
DB_PORT=3306
```

DB_USER=admin
DB_PASSWORD=password
DB_NAME=esms
JWT_SECRET=your_strong_secret_here
BUCKET_NAME=esms-bucket
BUCKET_REGION=us-east-1
ACCESS_KEY=AKIAYZZGS5NC2MJZPRIW
SECRET_ACCESS_KEY=qjAeT6gfGdTt+HQFZDu+je8ov9ysGbzNjrzWUhk3

Open/split separate terminal and enter directory for server
run the line:
npm install
npm run dev

the console should return success message when server is running
you can also see the server by going to localhost:5000

10.3 Instructions for System End Users

1. Login:
    a. To access the system, users should visit the designated domain URL (or localhost:3000 during testing) and enter their assigned username and password. Upon authentication, the system will automatically redirect them to a role-specific dashboard with tailored functionalities.
2. Admin:
    a. Administrators possess full system control, enabling them to manage all users, courses, and configurations. They can create, edit, or delete accounts for teachers, students, and parents, assign roles, and reset passwords. In the Course Management section, admins oversee course creation, teacher assignments, and student enrollments. They also configure system-wide settings, including themes, academic terms, and grading policies. Admins can access all gradebooks, attendance records, and reports, with privileges to modify any content or override settings as needed.
3. Teacher:
    a. Teachers are granted tools to manage their assigned courses, creating assignments, and setting deadlines. They record grades and feedback in the Gradebook, track student progress, and generate performance reports. Attendance features allow teachers to mark daily participation and submit exceptions. Communication tools enable announcements, discussion forums, and scheduling for parent-teacher conferences. Teachers also

monitor individual student records, flag academic concerns, and access submitted work.

4. Parent:
    a. Parents can view their child's academic progress through a dedicated portal. They access real-time grades, attendance records, and teacher feedback, with options to filter by course or timeframe. The system notifies parents of missed assignments, low grades, or behavioral flags. Communication features include messaging teachers, reviewing announcements, and scheduling meetings. Parents may also update their contact information and set notification preferences.
5. Student:
    a. Students interact with their personalized dashboard, which displays course schedules, assignments, and deadlines. They submit work electronically, view graded submissions with teacher feedback, and track overall performance. The attendance portal shows their participation records, while discussion forums facilitate peer and teacher collaboration. Students receive alerts for upcoming deadlines, grade updates, and school announcements, ensuring they stay informed.

11. Conclusion

11.1 Achievement

Our team successfully developed an Elementary School Management System that bridges critical gaps in parent-school communication while streamlining administrative workflows. Leveraging modern web technologies (React, Node.js, and MySQL), we delivered a secure platform with real-time grade updates, automated notifications, and role-based dashboards. Key milestones included implementing a dynamic gradebook for teachers, a parent portal with progress-tracking tools, and an admin panel for system-wide configuration—all validated through rigorous testing. This project demonstrates how technology can foster transparency in education while reducing administrative burdens, aligning perfectly with CSUSM's mission to drive innovation in community-focused solutions.

11.2 Lessons Learned

This project deepened our technical and collaborative skills:

- Technical Growth: We mastered full-stack development, from designing UML diagrams for system architecture to implementing secure API endpoints and responsive UI components. Challenges like optimizing database queries for enrollment data reinforced the importance of performance-centric design.
- User-Centric Design: Feedback from Professor Rahman and mock user testing revealed that simplicity is key—parents needed intuitive navigation, while teachers prioritized efficient data-entry workflows. This shaped our React-based interface with role-specific shortcuts.

- Team Dynamics: Agile methodologies kept us adaptable. Weekly standups with Professor Rahman helped us pivot smoothly when facing scope adjustments, such as adding real-time notifications mid-development.
- Future Considerations: Expanding mobile accessibility and integrating AI for predictive analytics (e.g., flagging at-risk students) emerged as promising enhancements for future iterations.

11.3 Acknowledgment

We would like to acknowledge this achievement to the university that provided us this opportunity, California State University San Marcos (CSUSM) and the CSTEM Department for providing resources that made this project possible. We also would pay respects and thanks to Professor Muhammad Lutfor Rahman, whose visionary guidance and unwavering support shaped every phase of this capstone—from problem identification to final implementation. His expertise in educational technology inspired our focus on parent engagement.and finally, to our families and mentors, whose encouragement sustained us through challenges, and whose financial support enabled our college journeys.

12. References

[1] React Library: https://reactjs.org/docs/getting-started.html

[2] Material UI Component Library: https://mui.com/material-ui/getting-started/

[3] Node.js Documentation: https://nodejs.org/en/docs/

[4] Express.js Documentation: https://expressjs.com

[5] MySQL Driver for Node.js: https://github.com/mysqljs/mysql

[6] AWS Elastic Beanstalk for Node.js: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_nodejs.html

[7] Visual Studio Code: https://code.visualstudio.com/docs

[8] Github for Elementary School Management System: https://github.com/Choshibear/CS490-StudentManagementSoftware

Table 4.1. Use Case Index Table

Project Name: Elementary School Student Management Software Proposal				
Use Case ID	**Use Case Name**	**Level**	**Author**	**Version**
UC-001	User Login and Authentication	Primary task	Priscilla Phung Tran	0.4
UC-002	Alert Notifications Management	Primary task	Priscilla Phung Tran	0.4
UC-003	Parent Views Child’s Academic Records & Progress	Primary task	Priscilla Phung Tran	0.5
UC-004	Announcements Management	Primary task	Priscilla Phung Tran	0.3
UC-005	Parent-Teacher Messaging	Primary task	Priscilla Phung Tran	0.2
UC-006	Attendance Tracking and Notification	Primary task	Priscilla Phung Tran	0.2
UC-007	Gradebook Management	Primary task	Priscilla Phung Tran	0.1
UC-008	Gradebook Viewing	Primary task	Priscilla Phung Tran	0.1
UC-009	Faculty can edit/update student records	Primary task	Farah Jaber	0.4
UC-010	Admin alters website asthetics	Subfunction	Farah Jaber	0.3
UC-011	Viewing Student Record	Primary task	Farah Jaber	0.3
Acknowledgment: Generated from the CapStone process management system ©2025				

Table 4.2. Use Case UC-001

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-001
Use Case Name:	User Login and Authentication
User Goal:	Allow users to securely log in to the system and access their appropriate role-based functionalities.
Scope:	System-wide authentication and role-based access
Level:	Primary task
Relevant User Reqs:	UF-D
Relevant System Reqs:	SF-D-01,SF-D-02
Primary Actor:	All users
Precondition:	The user must be registered in the system
Minimal Guarantee:	The system securely denies access if login credentials are incorrect
Success Guarantee:	The user is authenticated and granted access to the system with appropriate permissions
Trigger:	The user attempts to log in by entering their credentials on the login page
Success Scenario:	**Step** / **Actions** **1** The user navigates to the login page. **2** The user enters email/username and password. **3** The system verifies credentials **4** The system applies role based access control (RBAC) and identifies the user role **5** User is granted access and redirected to their appropriate dashboard
Extensions:	**Branching Scenarios**
3A	**Condition:** If the credentials are invalid, the system displays an error message and denies access
	Step / **Actions** **1** The system checks the entered credentials **2** If the credentials do not match any existing user, the system logs the failed attempt **3** The system displays an error message: "Invalid username or password. Please try again." **4** The user remains on the login page and must re-enter credentials
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.3. Use Case UC-002

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-002
Use Case Name:	Alert Notifications Management
User Goal:	Allow parents to subscribe or unsubscribe to specific alert notifications via email or SMS.
Scope:	Notification preference management
Level:	Primary task
Relevant User Reqs:	UF-S
Relevant System Reqs:	SF-S-01,SF-S-02
Primary Actor:	Parent
Precondition:	Parent is logged in
Minimal Guarantee:	Unsolicited alerts are not sent
Success Guarantee:	Parent’s notification preferences are updated and alerts are sent according to the selection
Trigger:	Parent accesses the Notification Settings interface
Success Scenario:	**Step** **Actions** **1** Parent logs in **2** Parent navigates to Notification Preferences **3** Parent selects or deselects specific alerts (attendance, grades, announcements, etc.) **4** System saves the updated preferences **5** Future alerts are sent based on these preferences
Extensions:	**Branching Scenarios**
4A	**Condition:** If preference update fails, the system displays an error message
	Step **Actions** **1** The system attempts to save the updated notification preferences **2** If the update process encounters an issue (e.g., connection failure), the system logs the failure **3** The system displays an error message: "Error" **4** The system retains the previous notification preferences **5** The user remains on the preference settings page and may attempt the update again
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.4. Use Case UC-003

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-003
Use Case Name:	Parent Views Child’s Academic Records & Progress
User Goal:	Allow parents with multiple children to select and access each child’s academic records and progress
Scope:	Viewing student records with multi-child managemen
Level:	Primary task
Relevant User Reqs:	UF-E
Relevant System Reqs:	SF-E-01,SF-E-02,SF-E-03,SF-F-03
Primary Actor:	Parent
Precondition:	Parent is logged in; Parent's account linked to one or more children; Student records exist in the system
Minimal Guarantee:	Unauthorized access is prevented
Success Guarantee:	Parent successfully selects a child and views that child’s academic records
Trigger:	Parent navigates to the Student Records section
Success Scenario:	**Step** / **Actions** **1** Parent logs in **2** Parent navigates to the Student Records dashboard **3** The system displays a child selection dashboard **4** Parent selects one child from the linked profiles **5** The system retrieves and displays the selected child’s academic records, enrollment history, and progress
Extensions:	**Branching Scenarios**
3A	**Condition:** If the child’s records are unavailable or an error occurs, the system displays an appropriate error message
	Step / **Actions** **1** The system attempts to fetch the student’s records from the database **2** If the records are missing, corrupted, or there is a database error, the system logs the issue **3** The system displays an error message: "Error" **4** The system does not load incomplete or incorrect data to prevent misinformation **5** The user remains on the Student Records page and may attempt to refresh or retry accessing the records
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.5. Use Case UC-004

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-004
Use Case Name:	Announcements Management
User Goal:	Enable admins or teachers to create and post group-specific announcements that are displayed on user dashboards
Scope:	Announcement posting and viewing
Level:	Primary task
Relevant User Reqs:	UF-R
Relevant System Reqs:	SF-R-01,SF-R-02,SF-R-03
Primary Actor:	Admin and Teacher
Precondition:	Admin or Teacher is logged in with announcement privileges
Minimal Guarantee:	Unauthorized posting is prevented
Success Guarantee:	Announcements are created, targeted, and displayed correctly on dashboards
Trigger:	Admin/Teacher initiates creation of an announcement
Success Scenario:	**Step** **Actions** **1** Admin/Teacher logs in and navigates to the Announcements section **2** They create a new announcement, specifying targeting criteria (grade, class, school-wide) **3** System saves the announcement and displays it on the relevant dashboards
Extensions:	**Branching Scenarios**
3A	**Condition:** If creation or display fails, the system displays an error message
	Step **Actions** **1** The system attempts to save or retrieve course content. **2** If an error occurs, the system logs it and displays an appropriate error message **3** If display fails, users can retry accessing the content **4** If creation fails, the system retains entered data when possible
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.6. Use Case UC-005

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-005
Use Case Name:	Parent-Teacher Messaging
User Goal:	Provide a secure, two-way messaging platform for communication between parents and teachers, including notifications for important updates
Scope:	Messaging and communication
Level:	Primary task
Relevant User Reqs:	UF-Q
Relevant System Reqs:	SF-Q-01,SF-Q-02
Primary Actor:	Parent/Teacher
Precondition:	Both parties are logged in and have access to the messaging interface
Minimal Guarantee:	Unauthorized messaging is prevented
Success Guarantee:	Messages are sent securely, received promptly, and notifications are issued
Trigger:	User composes and sends a message
Success Scenario:	**Step** **Actions** **1** Parent or teacher logs in and navigates to the Messaging section **2** User composes a new message, optionally attaching files **3** System sends the message securely and notifies the recipient **4** The conversation thread updates with the new message.
Extensions:	**Branching Scenarios**
3A	**Condition:** If sending fails, the system displays an error message
	Step **Actions** **1** The system attempts to send the message **2** If an error occurs, the system logs it and notifies the user with an error message **3** If sending fails, the system retains the unsent message when possible **4** The user can retry sending the message
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.7. Use Case UC-006

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-006
Use Case Name:	Attendance Tracking and Notification
User Goal:	Enable teachers to log attendance in real time and automatically notify parents of absences or tardiness
Scope:	Attendance tracking
Level:	Primary task
Relevant User Reqs:	UF-P
Relevant System Reqs:	SF-P-01,SF-P-02
Primary Actor:	Teacher/Parent
Precondition:	Teacher is logged in and has access to the class roster
Minimal Guarantee:	Attendance data is securely stored and modifications are restricted
Success Guarantee:	Attendance is logged correctly, and notifications are sent to parents automatically
Trigger:	Teacher marks attendance during class
Success Scenario:	**Step** **Actions** **1** Teacher logs in **2** Teacher navigates to the Attendance module **3** Teacher selects a class and marks attendance using the provided interface **4** System saves the attendance data **5** For marked absences or tardiness, the system sends notifications to parents via email/SMS
Extensions:	**Branching Scenarios**
3A	**Condition:** If data entry fails, the system displays an error message
	Step **Actions** **1** The system attempts to process the entered data **2** If an error occurs, the system logs it and notifies the user with an error message **3** The user is prompted to correct the data
5A	**Condition:** If notification fails, the system displays an error message
	Step **Actions** **1** The system attempts to notification **2** If an error occurs, the system logs it and notifies the user with an error message **3** The system provides an option to retry or notify an administrator
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.8. Use Case UC-007

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-007
Use Case Name:	Gradebook Management
User Goal:	Allow teachers to record, edit, and manage grades
Scope:	Gradebook management
Level:	Primary task
Relevant User Reqs:	UF-I
Relevant System Reqs:	SF-I-01,SF-I-02,SF-I-03
Primary Actor:	Teacher
Precondition:	Teacher is logged in with access to the Gradebook module
Minimal Guarantee:	Unauthorized modifications are prevented
Success Guarantee:	Grades are recorded, calculatedcorectly
Trigger:	Teacher navigates to the Gradebook section
Success Scenario:	**Step** **Actions** **1** Teacher logs in **2** Teacher navigates to Gradebook **3** Teacher records or edits grades for assignments, tests, or projects **4** System calculates overall grades and updates records
Extensions:	**Branching Scenarios**
4A	**Condition:**
	Step **Actions** **1** When a teacher enters or edits a grade, the system attempts to record the data and calculate the overall grade **2** If either the grade entry or the calculation process fails (due to invalid input, database error, or calculation error), the system logs the error **3** The system then displays an error message **4** The teacher is prompted to re-enter the grade

Acknowledgment: Generated from the CapStone process management system ©2025

Table 4.9. Use Case UC-008

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-008
Use Case Name:	Gradebook Viewing
User Goal:	Allow parents (and students) to view grades using various grading scales
Scope:	Gradebook Viewing
Level:	Primary task
Relevant User Reqs:	UF-I
Relevant System Reqs:	SF-I-04
Primary Actor:	Parent/Student
Precondition:	Parent is logged in with access to the Gradebook module; Grade data exists in the system.
Minimal Guarantee:	Unauthorized access is prevented
Success Guarantee:	Grades are displayed correctly
Trigger:	Parent navigates to the Gradebook section
Success Scenario:	**Step** **Actions** **1** Parent or student logs in **2** They view the Gradebook in a read-only format
Extensions:	**Branching Scenarios**
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.10. Use Case UC-009

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-009
Use Case Name:	Faculty can edit/update student records
User Goal:	Allow authorized users (Faculty) to store and update student records
Scope:	Student Records Management
Level:	Primary task
Relevant User Reqs:	UF-F
Relevant System Reqs:	SF-F-01,SF-F-03
Primary Actor:	Faculty
Precondition:	User is authenticated and has appropriate access rights
Minimal Guarantee:	The system prevents unauthorized changes and data corruption
Success Guarantee:	Student records are stored and retrieved correctly
Trigger:	The user accesses the Student Records module
Success Scenario:	**Step** **Actions** **1** The user logs in **2** The user navigates to the Student Records section **3** The user updates student information (personal details, enrollment, progress) **4** The system validates and securely stores the updates
Extensions:	**Branching Scenarios**
3A	**Condition:** If an invalid input is detected, an error message is displayed
	Step **Actions** **1** The system validates the input against predefined constraints **2** The system notifies the user: "Invalid input detected. Please check the format and required fields." **3** The user remains on the input page and is prompted to re-enter the correct data
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.11. Use Case UC-010

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-010
Use Case Name:	Admin alters website asthetics
User Goal:	Allow admins to define and modify student record fields and aesthetics.
Scope:	Student Records Management
Level:	Subfunction
Relevant User Reqs:	UF-F
Relevant System Reqs:	SF-F-02
Primary Actor:	Admin
Precondition:	Admin is logged into the system
Minimal Guarantee:	The system prevents misconfiguration, ensuring required fields remain intact
Success Guarantee:	Admins can successfully modify record fields and aesthetics, and changes are reflected in the system.
Trigger:	Admin navigates to the Customization page
Success Scenario:	**Step** **Actions** **1** The user navigates to the Customization settings **2** The user selects customization options (e.g., colors, layout, additional fields). **3** The user confirms and saves changes.
Extensions:	**Branching Scenarios**
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.12. Use Case UC-011

Project Name:	**Elementary School Student Management Software Proposal**
Use Case ID:	UC-011
Use Case Name:	Viewing Student Record
User Goal:	Allow parents and students to securely view student records, including personal details, enrollment history, and academic progress.
Scope:	Student Records Management System
Level:	Primary task
Relevant User Reqs:	UF-E,UF-F
Relevant System Reqs:	SF-F-01
Primary Actor:	Parents and Student
Precondition:	User is logged in
Minimal Guarantee:	Unauthorized access is denied, and data integrity is maintained
Success Guarantee:	Users can securely view their own (or their child's) student records
Trigger:	The user navigates to the Student Records section
Success Scenario:	**Step** **Actions** **1** The user navigates to the Student Records section. **2** The user selects a student profile (if parent has more than1 child). **3** The user views details like grades, attendance, and progress
Extensions:	**Branching Scenarios**
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.15. User Functional Requirements: UF-D

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UF-D**			**Type**	**Functional** / **Non-Functional**
Creation:	**Feb 27 2025 09:36 AM**			**User**	☒ / ☐
Modification:	**Mar 06 2025 12:12 AM**			**System**	☐ / ☐
Description:	Users must be able to log into application and gain access to authorized information and functionalities based on their assigned user role (Admin, Teacher, Parent, Student).				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SF-D-01, SF-D-02**				
Justify why UF-D can be completely covered by SF-D-01, SF-D-02	To be added later				
Traceability:	**Use cases cf.**	**UC-001**			
	Test cases cf.	**TC-002**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.16. User Functional Requirements: UF-E

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UF-E**			**Type**	**Functional** / **Non-Functional**
Creation:	**Feb 27 2025 09:40 AM**			**User**	☒ / ☐
Modification:	**Mar 06 2025 12:16 AM**			**System**	☐ / ☐
Description:	Parents with multiple children should be able to select and access each child’s information separately.				
Priority:	Highest	High	**✔ Medium**	Low	Lowest
This Req. is Refined Into:	**SF-E-01, SF-E-02, SF-E-03**				
Justify why UF-E can be completely covered by SF-E-01, SF-E-02, SF-E-03	To be added later				
Traceability:	**Use cases cf.**	**UC-003, UC-011**			
	Test cases cf.	**TC-003**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.17. User Functional Requirements: UF-F

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UF-F**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Feb 27 2025 09:44 AM**				
Modification:	**Mar 06 2025 12:19 AM**			**User** ☒ / ☐	
				System ☐ / ☐	
Description:	Student Records will be used to store student details, enrollment history and academic progress, allowing admins to customize data and aesthetics relevant to school, teachers to maintain information and parents/students to view records.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Refined Into:	**SF-F-01, SF-F-02, SF-F-03**				
Justify why UF-F can be completely covered by SF-F-01, SF-F-02, SF-F-03	To be added later				
Traceability:	**Use cases cf.**	**UC-009, UC-010, UC-011**			
	Test cases cf.	**TC-004**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.18. User Functional Requirements: UF-H

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UF-H**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Feb 27 2025 09:47 AM**				
Modification:	**Mar 06 2025 12:43 AM**			**User** ☒ / ☐	
				System ☐ / ☐	
Description:	Using a Course Management feature, teachers will be able to create, manage, and organize and parents/students will be able to view course materials of varying content and media types.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Refined Into:	**SF-H-01, SF-H-02, SF-H-03, SF-H-04**				
Justify why UF-H can be completely covered by SF-H-01, SF-H-02, SF-H-03, SF-H-04	To be added later				
Traceability:	**Use cases cf.**	**UC-012, UC-013**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.19. User Functional Requirements: UF-I

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	UF-I			Type: Functional / Non-Functional	
Creation:	Feb 27 2025 09:47 AM			User: ☒ / ☐	
Modification:	Mar 06 2025 12:40 AM			System: ☐ / ☐	
Description:	A Gradebook will allow teachers to record and manage, and allow parents to view grades for assignments, tests, and projects with support for various grading scales.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Refined Into:	SF-I-01, SF-I-02, SF-I-03, SF-I-04				
Justify why UF-I can be completely covered by SF-I-01, SF-I-02, SF-I-03, SF-I-04	To be added later				
Traceability:	Use cases cf.	UC-007, UC-008			
	Test cases cf.	TC-007			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.20. User Functional Requirements: UF-P

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	UF-P			Type: Functional / Non-Functional	
Creation:	Feb 27 2025 11:56 AM			User: ☒ / ☐	
Modification:	Mar 06 2025 12:48 AM			System: ☐ / ☐	
Description:	An Attendance Tracking feature will enable teachers to log attendance in real-time and automatically notify parents about absences or tardiness.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Refined Into:	SF-P-01, SF-P-02				
Justify why UF-P can be completely covered by SF-P-01, SF-P-02	To be added later				
Traceability:	Use cases cf.	UC-006			
	Test cases cf.	TC-006			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.21. User Functional Requirements: UF-Q

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	UF-Q			Type: Functional / Non-Functional	
Creation:	Feb 27 2025 11:56 AM			User: ☒ / ☐	
Modification:	Mar 06 2025 12:55 AM			System: ☐ / ☐	
Description:	A Parent-Teacher Communication feature must provide a two-way messaging system for personal communication and notifications for important updates.				
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Refined Into:	SF-Q-01, SF-Q-02				
Justify why UF-Q can be completely covered by SF-Q-01, SF-Q-02	To be added later				
Traceability:	Use cases cf.	UC-005			
	Test cases cf.	TC-005			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.22. User Functional Requirements: UF-R

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	UF-R			Type: Functional / Non-Functional	
Creation:	Feb 27 2025 11:57 AM			User: ☒ / ☐	
Modification:	Mar 06 2025 12:57 AM			System: ☐ / ☐	
Description:	An Announcement feature should enable admins or teachers to post group specific custom announcements to be displayed in a dashboard.				
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Refined Into:	SF-R-01, SF-R-02, SF-R-03				
Justify why UF-R can be completely covered by SF-R-01, SF-R-02, SF-R-03	To be added later				
Traceability:	Use cases cf.	UC-004			
	Test cases cf.	TC-008			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.23. User Functional Requirements: UF-S

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UF-S**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Feb 27 2025 11:58 AM**			**User** ☒	☐
Modification:	**Mar 06 2025 01:00 AM**			**System** ☐	☐
Description:	Parents should be able to subscribe and unsubscribe to selected Alert Notifications via email or SMS.				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Refined Into:	**SF-S-01, SF-S-02**				
Justify why UF-S can be completely covered by SF-S-01, SF-S-02	To be added later				
Traceability:	**Use cases cf.**	**UC-002**			
	Test cases cf.	**TC-009**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.24. User NonFunctional Requirements: UP-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-02**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Feb 13 2025 01:16 AM**			**User** ☐	☒
Modification:	**Mar 06 2025 03:44 AM**			**System** ☐	☐
Description:	Comprehensive user support should be available.			**Product (sub-type below)**	
				Usability Requirements	
Priority:	Highest	High	Medium	✔ **Low**	Lowest
This Req. is Refined Into:	**SP-02-01, SP-02-02, SP-02-03**				
Justify why UP-02 can be completely covered by SP-02-01, SP-02-02, SP-02-03	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.25. User NonFunctional Requirements: UP-05

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-05**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Feb 27 2025 11:57 AM**			**User** ☐ ☒	
Modification:	**Mar 06 2025 03:37 AM**			**System** ☐ ☐	
Description:	The system must have an intuitive, user-friendly interface whose commonly used features are easily accessed.			**Product (sub-type below)**	
				Usability Requirements	
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-05-01, SP-05-02, SP-05-03, SP-05-04**				
Justify why UP-05 can be completely covered by SP-05-01, SP-05-02, SP-05-03, SP-05-04	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.26. User NonFunctional Requirements: UP-07

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-07**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Mar 06 2025 03:46 AM**			**User** ☐ ☒	
Modification:	**Mar 06 2025 03:46 AM**			**System** ☐ ☐	
Description:	The system must be compatible with modern web browsers and mobile devices, ensuring a responsive design for smartphones and tablets.			**Product (sub-type below)**	
				Usability Requirements	
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-07-01, SP-07-02, SP-07-03**				
Justify why UP-07 can be completely covered by SP-07-01, SP-07-02, SP-07-03	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.27. User NonFunctional Requirements: UP-06

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-06**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Mar 04 2025 01:21 PM**			**User** ☐ ☒	
Modification:	**Mar 06 2025 03:50 AM**			**System** ☐ ☐	
Description:	Real-time updates (e.g., attendance, notifications) should occur with minimal delay.			**Product (sub-type below)**	
				Performance Requirements	
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-06-01, SP-06-02, SP-06-03**				
Justify why UP-06 can be completely covered by SP-06-01, SP-06-02, SP-06-03	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.28. User NonFunctional Requirements: UP-08

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-08**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Mar 06 2025 03:53 AM**			**User** ☐ ☒	
Modification:	**Mar 06 2025 03:54 AM**			**System** ☐ ☐	
Description:	The system must handle up to 1000 concurrent users without performance degradation.			**Product (sub-type below)**	
				Performance Requirements	
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-08-01, SP-08-02**				
Justify why UP-08 can be completely covered by SP-08-01, SP-08-02	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.29. User NonFunctional Requirements: UP-09

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-09**				
Creation:	**Mar 06 2025 03:55 AM**				
Modification:	**Mar 06 2025 03:56 AM**				
Description:	The system must be scalable, supporting additional schools, grades, and users without significant reconfiguration.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-09-01, SP-09-02, SP-09-03**				
Justify why UP-09 can be completely covered by SP-09-01, SP-09-02, SP-09-03	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☒
System	☐	☐

Product (sub-type below)

Performance Requirements

Table 4.30. User NonFunctional Requirements: UP-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-01**				
Creation:	**Feb 12 2025 11:05 PM**				
Modification:	**Mar 06 2025 03:25 AM**				
Description:	All data must be encrypted during storage and transmission.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-01-01, SP-01-02**				
Justify why UP-01 can be completely covered by SP-01-01, SP-01-02	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**TC-001**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☒
System	☐	☐

Product (sub-type below)

Availability/Reliability/Security

Table 4.31. User NonFunctional Requirements: UP-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-03**				
Creation:	**Feb 27 2025 09:43 AM**				
Modification:	**Mar 06 2025 03:28 AM**				
Description:	Role-Based Access Control (RBAC) must ensure users access only authorized data.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SP-03-01, SP-03-02**				
Justify why UP-03 can be completely covered by SP-03-01, SP-03-02	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☒
System	☐	☐

Product (sub-type below)

Availability/Reliability/Security

Table 4.32. User NonFunctional Requirements: UP-04

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UP-04**				
Creation:	**Mar 06 2025 03:33 AM**				
Modification:	**Mar 06 2025 03:33 AM**				
Description:	The system must maintain 99.9% uptime, with automatic backups and a fast recovery time in case of failure.				
Priority:	Highest	High	**✔ Medium**	Low	Lowest
This Req. is Refined Into:	**SP-04-01, SP-04-02, SP-04-03**				
Justify why UP-04 can be completely covered by SP-04-01, SP-04-02, SP-04-03	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☒
System	☐	☐

Product (sub-type below)

Availability/Reliability/Security

Table 4.33. User NonFunctional Requirements: UE-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**UE-01**				
Creation:	**Mar 06 2025 10:43 AM**				
Modification:	**Mar 06 2025 10:44 AM**				
Description:	System must securely store all personal information, sensitive or otherwise.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:	**SE-01-01**				
Justify why UE-01 can be completely covered by SE-01-01	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**TC-001**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☒
System	☐	☐

External (sub-type below)

Legislative Requirements on Safety/Security

Table 4.34. System Functional Requirements: SF-D-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-D-01**				
Creation:	**Feb 27 2025 01:34 PM**				
Modification:	**Mar 06 2025 12:14 AM**				
Description:	The system shall authenticate users using email/username and password before granting access.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-D**				
Justify why meeting SF-D-01 can contribute to the fulfilment of UF-D	Authentication ensures that only valid users can access the system, preventing unauthorized access.				
Traceability:	**Use cases cf.**	**UC-001**			
	Test cases cf.	**TC-002**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☒	☐

Table 4.35. System Functional Requirements: SF-D-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-D-02**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Mar 06 2025 12:14 AM**			**User** ☐ ☐	
Modification:	**Mar 06 2025 12:15 AM**			**System** ☒ ☐	
Description:	The system shall implement role-based access control (RBAC) to restrict access based on user roles.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UF-D**				
Justify why meeting SF-D-02 can contribute to the fulfilment of UF-D	RBAC ensures that each user type (Admin, Teacher, Parent, Student) has access only to the functionalities and data relevant to their role. This prevents unauthorized modifications and maintains data integrity by ensuring that, for example, students cannot edit grades or attendance records.				
Traceability:	**Use cases cf.**	**UC-001**			
	Test cases cf.	**TC-002**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.36. System Functional Requirements: SF-E-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-E-01**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Feb 27 2025 01:45 PM**			**User** ☐ ☐	
Modification:	**Mar 06 2025 12:16 AM**			**System** ☒ ☐	
Description:	The system shall allow a parent to be linked to multiple children.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UF-E**				
Justify why meeting SF-E-01 can contribute to the fulfilment of UF-E	By associating a parent account with multiple child profiles, the system ensures that parents can access all their children’s academic records within a single login session.				
Traceability:	**Use cases cf.**	**UC-003**			
	Test cases cf.	**TC-003**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.37. System Functional Requirements: SF-E-02

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-E-02			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 12:16 AM			User ☐ / ☐	
Modification:	Mar 06 2025 12:17 AM			System ☒ / ☐	
Description:	The system shall provide a child selection dashboard for parents to switch between child profiles.				
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-02 can contribute to the fulfilment of UF-E	The dashboard provides an intuitive way for parents to select a child and view their records, preventing confusion and improving user experience.				
Traceability:	Use cases cf.	UC-003			
	Test cases cf.	TC-003			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.38. System Functional Requirements: SF-E-03

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-E-03			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 12:17 AM			User ☐ / ☐	
Modification:	Mar 06 2025 12:18 AM			System ☒ / ☐	
Description:	The system shall restrict access to only the children associated with the parent’s account.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	UF-E				
Justify why meeting SF-E-03 can contribute to the fulfilment of UF-E	This ensures that a parent cannot access records of students they are not authorized to view.				
Traceability:	Use cases cf.	UC-003			
	Test cases cf.	TC-003			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.39. System Functional Requirements: SF-F-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-F-01			Type / Functional / Non-Functional	
Creation:	Feb 27 2025 03:00 PM				
Modification:	Mar 06 2025 12:19 AM			User ☐ / ☐	
				System ☒ / ☐	
Description:	The system shall securely store student records, including personal details, enrollment history, and academic progress in a centralized database.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-F				
Justify why meeting SF-F-01 can contribute to the fulfilment of UF-F	A centralized database ensures that student information is well-organized, accessible, and secure, preventing data loss and ensuring easy retrieval of records when needed.				
Traceability:	Use cases cf.	UC-009, UC-011			
	Test cases cf.	TC-004			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.40. System Functional Requirements: SF-F-02

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-F-02			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 12:20 AM				
Modification:	Mar 06 2025 12:21 AM			User ☐ / ☐	
				System ☒ / ☐	
Description:	The system shall allow admins to define and customize data fields and aesthetics (e.g., layout or additional custom fields) via an administrative interface.				
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:	UF-F				
Justify why meeting SF-F-02 can contribute to the fulfilment of UF-F	Customization allows schools to capture additional information based on institutional requirements and present it in a format that aligns with their internal policies.				
Traceability:	Use cases cf.	UC-010			
	Test cases cf.	TC-004			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.41. System Functional Requirements: SF-F-03

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-F-03				
Creation:	Mar 06 2025 12:21 AM				
Modification:	Mar 06 2025 12:21 AM				
Description:	Teachers shall have write access to update student records, while parents and students shall have read-only access.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-F				
Justify why meeting SF-F-03 can contribute to the fulfilment of UF-F	Role-based data access ensures that only authorized personnel (teachers) can modify student records while parents and students can view but not alter the records, maintaining data integrity.				
Traceability:	Use cases cf.	UC-003, UC-009			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Type	Functional	Non-Functional
User	☐	☐
System	☒	☐

Table 4.42. System Functional Requirements: SF-H-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-H-01				
Creation:	Feb 27 2025 03:13 PM				
Modification:	Mar 06 2025 12:45 AM				
Description:	The system shall allow teachers to create, manage, and organize course content (lessons, assignments, assessments, multimedia files) by subject, grade, and module.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-H				
Justify why meeting SF-H-01 can contribute to the fulfilment of UF-H	Teachers need a structured way to manage course materials, ensuring that students receive organized and accessible content.				
Traceability:	Use cases cf.	UC-012			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Type	Functional	Non-Functional
User	☐	☐
System	☒	☐

Table 4.43. System Functional Requirements: SF-H-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-H-02**				
Creation:	**Mar 06 2025 12:33 AM**				
Modification:	**Mar 06 2025 12:45 AM**				
Description:	The system shall support various content types including documents, videos, and interactive elements (e.g., quizzes).				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-H**				
Justify why meeting SF-H-02 can contribute to the fulfilment of UF-H	Different subjects require diverse teaching materials, and supporting multiple formats enhances learning experiences.				
Traceability:	**Use cases cf.**	**UC-012, UC-013**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☒	☐

Table 4.44. System Functional Requirements: SF-H-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-H-03**				
Creation:	**Mar 06 2025 12:34 AM**				
Modification:	**Mar 06 2025 12:46 AM**				
Description:	The system shall allow teachers to set deadlines for assignments.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-H**				
Justify why meeting SF-H-03 can contribute to the fulfilment of UF-H	Assignments with clear deadlines help students manage their workload and ensure timely submission.				
Traceability:	**Use cases cf.**	**UC-012**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☒	☐

Table 4.45. System Functional Requirements: SF-H-04

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-H-04			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 12:34 AM				
Modification:	Mar 06 2025 12:46 AM			User ☐ / ☐	
				System ☒ / ☐	
Description:	The system shall automatically display these materials in the student and parent portals.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	UF-H				
Justify why meeting SF-H-04 can contribute to the fulfilment of UF-H	Automatic display ensures that students and parents can access relevant course materials without requiring additional manual intervention from teachers.				
Traceability:	Use cases cf.	UC-013			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.46. System Functional Requirements: SF-I-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-I-01			Type / Functional / Non-Functional	
Creation:	Feb 27 2025 03:22 PM				
Modification:	Mar 06 2025 12:39 AM			User ☐ / ☐	
				System ☒ / ☐	
Description:	The system shall allow teachers to record, edit, and manage grades for assignments, tests, and projects.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-I				
Justify why meeting SF-I-01 can contribute to the fulfilment of UF-I	Teachers need a structured system to input and manage student grades, ensuring accuracy and easy updates when necessary.				
Traceability:	Use cases cf.	UC-007			
	Test cases cf.	TC-007			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.47. System Functional Requirements: SF-I-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-I-02**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Mar 06 2025 12:40 AM**			**User** ☐ ☐	
Modification:	**Mar 06 2025 12:41 AM**			**System** ☒ ☐	
Description:	The system shall support multiple grading scales (letter, numeric, pass/fail).				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-I**				
Justify why meeting SF-I-02 can contribute to the fulfilment of UF-I	Different schools and courses may use different grading systems, and supporting multiple scales ensures flexibility in assessment.				
Traceability:	**Use cases cf.**	**UC-007**			
	Test cases cf.	**TC-007**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.48. System Functional Requirements: SF-I-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-I-03**			**Type** / **Functional** / **Non-Functional**	
Creation:	**Mar 06 2025 12:41 AM**			**User** ☐ ☐	
Modification:	**Mar 06 2025 12:41 AM**			**System** ☒ ☐	
Description:	The system shall calculate overall grades based on weighted assessments.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-I**				
Justify why meeting SF-I-03 can contribute to the fulfilment of UF-I	Automating grade calculations reduces manual errors and ensures that students' progress is assessed consistently based on predefined weightings.				
Traceability:	**Use cases cf.**	**UC-007**			
	Test cases cf.	**TC-007**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.49. System Functional Requirements: SF-I-04

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-I-04				
Creation:	Mar 06 2025 12:42 AM				
Modification:	Mar 06 2025 12:42 AM				
Description:	The system shall ensure that parents and students have read-only access to grade information.				
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UF-I				
Justify why meeting SF-I-04 can contribute to the fulfilment of UF-I	Read-only access prevents unauthorized changes while still allowing parents and students to monitor academic progress.				
Traceability:	Use cases cf.	UC-008			
	Test cases cf.	TC-007			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Type	Functional	Non-Functional
User	☐	☐
System	☒	☐

Table 4.50. System Functional Requirements: SF-P-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-P-01				
Creation:	Mar 04 2025 01:25 PM				
Modification:	Mar 06 2025 12:50 AM				
Description:	The system shall automatically notify parents via email/SMS when a student is absent or tardy.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	UF-P				
Justify why meeting SF-P-01 can contribute to the fulfilment of UF-P	Immediate notifications keep parents informed of their child's attendance status, enabling timely interventions for absences or tardiness.				
Traceability:	Use cases cf.	UC-006			
	Test cases cf.	TC-006			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Type	Functional	Non-Functional
User	☐	☐
System	☒	☐

Table 4.51. System Functional Requirements: SF-P-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-P-02**			**Type**	**Functional** / **Non-Functional**
Creation:	**Mar 06 2025 12:50 AM**			**User**	☐ / ☐
Modification:	**Mar 06 2025 12:51 AM**			**System**	☒ / ☐
Description:	The system shall provide an interface (using checkboxes or dropdowns) for teachers to mark student attendance in real time.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-P**				
Justify why meeting SF-P-02 can contribute to the fulfilment of UF-P	A simple, user-friendly interface ensures that teachers can quickly and efficiently log attendance during class, minimizing disruptions.				
Traceability:	**Use cases cf.**	**UC-006**			
	Test cases cf.	**TC-006**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.52. System Functional Requirements: SF-Q-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-Q-01**			**Type**	**Functional** / **Non-Functional**
Creation:	**Mar 04 2025 01:00 PM**			**User**	☐ / ☐
Modification:	**Mar 06 2025 12:56 AM**			**System**	☒ / ☐
Description:	The system shall provide a secure, two-way messaging platform for communication between parents and teachers, including support for text messages and file attachments.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UF-Q**				
Justify why meeting SF-Q-01 can contribute to the fulfilment of UF-Q	Secure communication allows for effective collaboration between parents and teachers, ensuring privacy and clarity in student-related discussions.				
Traceability:	**Use cases cf.**	**UC-005**			
	Test cases cf.	**TC-005**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.53. System Functional Requirements: SF-Q-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-Q-02**				
Creation:	**Mar 06 2025 12:56 AM**				
Modification:	**Mar 06 2025 12:57 AM**				
Description:	The system shall notify users (via email or SMS) when a new message is received				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UF-Q**				
Justify why meeting SF-Q-02 can contribute to the fulfilment of UF-Q	Notifications ensure that important messages are promptly seen and addressed, preventing missed communications.				
Traceability:	**Use cases cf.**	**UC-005**			
	Test cases cf.	**TC-005**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☒	☐

Table 4.54. System Functional Requirements: SF-R-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-R-01**				
Creation:	**Mar 04 2025 01:08 PM**				
Modification:	**Mar 06 2025 12:58 AM**				
Description:	The system shall allow admins/teachers to create announcements.				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UF-R**				
Justify why meeting SF-R-01 can contribute to the fulfilment of UF-R	Teachers and administrators need a way to share important information with students and parents efficiently.				
Traceability:	**Use cases cf.**	**UC-004**			
	Test cases cf.	**TC-008**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☒	☐

Table 4.55. System Functional Requirements: SF-R-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-R-02**			**Type**	**Functional** / **Non-Functional**
Creation:	**Mar 06 2025 12:58 AM**			**User**	☐ / ☐
Modification:	**Mar 06 2025 12:59 AM**			**System**	☒ / ☐
Description:	The system shall allow announcements to be targeted by grade, class, or school-wide.				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UF-R**				
Justify why meeting SF-R-02 can contribute to the fulfilment of UF-R	Targeting announcements ensures that only relevant users receive specific information, preventing unnecessary notifications.				
Traceability:	**Use cases cf.**	**UC-004**			
	Test cases cf.	**TC-008**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.56. System Functional Requirements: SF-R-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SF-R-03**			**Type**	**Functional** / **Non-Functional**
Creation:	**Mar 06 2025 12:59 AM**			**User**	☐ / ☐
Modification:	**Mar 06 2025 12:59 AM**			**System**	☒ / ☐
Description:	Announcements shall be displayed on the user dashboard.				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UF-R**				
Justify why meeting SF-R-03 can contribute to the fulfilment of UF-R	Displaying announcements on the dashboard ensures easy access and visibility for all authorized users.				
Traceability:	**Use cases cf.**	**UC-004**			
	Test cases cf.	**TC-008**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.57. System Functional Requirements: SF-S-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-S-01			Type / Functional / Non-Functional	
Creation:	Mar 04 2025 01:11 PM				
Modification:	Mar 06 2025 01:01 AM			User ☐ ☐	
				System ☒ ☐	
Description:	The system shall allow parents to enable or disable specific notifications (e.g., attendance, grades, announcements) through a user preference interface.				
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:	UF-S				
Justify why meeting SF-S-01 can contribute to the fulfilment of UF-S	Providing customizable notification settings ensures that parents receive only the alerts that are most relevant to them, improving user experience.				
Traceability:	Use cases cf.	UC-002			
	Test cases cf.	TC-009			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.58. System Functional Requirements: SF-S-02

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SF-S-02			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 01:01 AM				
Modification:	Mar 06 2025 01:01 AM			User ☐ ☐	
				System ☒ ☐	
Description:	The system shall send alerts via email and SMS based on user preferences.				
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:	UF-S				
Justify why meeting SF-S-02 can contribute to the fulfilment of UF-S	Multiple delivery methods ensure that parents receive notifications through their preferred communication channel, increasing engagement and awareness.				
Traceability:	Use cases cf.	UC-002			
	Test cases cf.	TC-009			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.59. System NonFunctional Requirements: SP-02-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-02-01**				
Creation:	**Mar 04 2025 01:38 PM**				
Modification:	**Mar 06 2025 03:44 AM**				
Description:	The system must provide user documentation including tooltips, FAQs, and step-by-step guides available from an in-app help center.				
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Engineered From:	**UP-02**				
Justify why meeting SP-02-01 can contribute to the fulfilment of UP-02	Reduces dependency on external support by allowing users to find answers independently.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)
Usability Requirements

Table 4.60. System NonFunctional Requirements: SP-02-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-02-02**				
Creation:	**Mar 04 2025 01:42 PM**				
Modification:	**Mar 06 2025 03:45 AM**				
Description:	System must provide an interactive website tutorial or inline guidance for key features to assist first-time users.				
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Engineered From:	**UP-02**				
Justify why meeting SP-02-02 can contribute to the fulfilment of UP-02	Reduces the learning curve, allowing users to adapt to the system quickly and effectively.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)
Usability Requirements

Table 4.61. System NonFunctional Requirements: SP-02-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-02-03**				
Creation:	**Mar 04 2025 01:48 PM**				
Modification:	**Mar 04 2025 01:51 PM**				
Description:	System should be designed with a modular architecture				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UP-02**				
Justify why meeting SP-02-03 can contribute to the fulfilment of UP-02	This will ensure reliability of system throughout maintenance and scalability				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Usability Requirements

Table 4.62. System NonFunctional Requirements: SP-05-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-05-01**				
Creation:	**Mar 04 2025 01:43 PM**				
Modification:	**Mar 06 2025 03:40 AM**				
Description:	The system must have a clean, consistent design with a clear layout to facilitate navigation.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-05**				
Justify why meeting SP-05-01 can contribute to the fulfilment of UP-05	Ensures the application is clear and easy to read.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Usability Requirements

Table 4.63. System NonFunctional Requirements: SP-05-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-05-02**				
Creation:	**Mar 04 2025 01:44 PM**				
Modification:	**Mar 06 2025 03:41 AM**				
Description:	Frequently accessed features should be accessible with one click from a menu anchored to every page.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-05**				
Justify why meeting SP-05-02 can contribute to the fulfilment of UP-05	Reduces time spent navigating the system, enhancing user efficiency and overall experience.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Usability Requirements

Table 4.64. System NonFunctional Requirements: SP-05-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-05-03**				
Creation:	**Mar 06 2025 03:42 AM**				
Modification:	**Mar 06 2025 03:42 AM**				
Description:	The system must maintain a consistent and responsive UI across all pages with a standardized design language.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-05**				
Justify why meeting SP-05-03 can contribute to the fulfilment of UP-05	Enhances usability by ensuring familiarity and reducing cognitive load for users.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Usability Requirements

Table 4.65. System NonFunctional Requirements: SP-05-04

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-05-04**				
Creation:	**Mar 06 2025 03:42 AM**				
Modification:	**Mar 06 2025 03:43 AM**				
Description:	Icons, labels, and tooltips must be used effectively to reduce reliance on text-heavy navigation.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-05**				
Justify why meeting SP-05-04 can contribute to the fulfilment of UP-05	Helps new users understand system functions quickly without needing extensive training.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Usability Requirements

Table 4.66. System NonFunctional Requirements: SP-07-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-07-01**				
Creation:	**Mar 06 2025 03:46 AM**				
Modification:	**Mar 06 2025 03:47 AM**				
Description:	The system must be compatible with Chrome, Firefox, Safari, and Edge, and should maintain usability across desktop and mobile devices.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-07**				
Justify why meeting SP-07-01 can contribute to the fulfilment of UP-07	Ensures the system remains functional across widely used browsers, preventing access issues.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Usability Requirements

Table 4.67. System NonFunctional Requirements: SP-07-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-07-02**				
Creation:	**Mar 06 2025 03:47 AM**				
Modification:	**Mar 06 2025 03:48 AM**				
Description:	The system must be mobile-responsive, automatically adjusting UI elements based on screen size.				
Priority:	Highest	High	**✔ Medium**	Low	Lowest
This Req. is Engineered From:	**UP-07**				
Justify why meeting SP-07-02 can contribute to the fulfilment of UP-07	Provides an optimal experience for users accessing the system on different devices.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)
Usability Requirements

Table 4.68. System NonFunctional Requirements: SP-07-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-07-03**				
Creation:	**Mar 06 2025 03:48 AM**				
Modification:	**Mar 06 2025 03:48 AM**				
Description:	All features must be accessible on both desktop (≥1024px screen width) and mobile (≥5 inches screen width) without requiring a separate app.				
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Engineered From:	**UP-07**				
Justify why meeting SP-07-03 can contribute to the fulfilment of UP-07	Reduces development and maintenance costs while ensuring accessibility on all platforms.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)
Usability Requirements

Table 4.69. System NonFunctional Requirements: SP-06-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-06-01**				
Creation:	**Mar 04 2025 03:03 PM**				
Modification:	**Mar 06 2025 03:51 AM**				
Description:	The system must use WebSockets or push notifications to enable real-time updates.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-06**				
Justify why meeting SP-06-01 can contribute to the fulfilment of UP-06	Reduces latency, providing instant updates without user refresh.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Performance Requirements

Table 4.70. System NonFunctional Requirements: SP-06-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-06-02**				
Creation:	**Mar 06 2025 03:52 AM**				
Modification:	**Mar 06 2025 03:52 AM**				
Description:	The notification system must allow asynchronous processing to prevent bottlenecks.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-06**				
Justify why meeting SP-06-02 can contribute to the fulfilment of UP-06	Ensures smooth operation without delaying other system processes.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Performance Requirements

Table 4.71. System NonFunctional Requirements: SP-06-03

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SP-06-03			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 03:52 AM				
Modification:	Mar 06 2025 03:53 AM			User ☐ ☐	
				System ☐ ☒	
Description:	Attendance logs, grade updates, and notifications must be processed within 2 seconds.			Product (sub-type below)	
				Performance Requirements	
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UP-06				
Justify why meeting SP-06-03 can contribute to the fulfilment of UP-06	Ensures timely communication of critical student-related information.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	TC-006			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.72. System NonFunctional Requirements: SP-08-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SP-08-01			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 03:54 AM				
Modification:	Mar 06 2025 03:54 AM			User ☐ ☐	
				System ☐ ☒	
Description:	The system must support 1000 concurrent users while maintaining response times below 3 seconds.			Product (sub-type below)	
				Performance Requirements	
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UP-08				
Justify why meeting SP-08-01 can contribute to the fulfilment of UP-08	Ensures smooth system performance even during peak usage periods.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.73. System NonFunctional Requirements: SP-08-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-08-02**				
Creation:	**Mar 06 2025 03:54 AM**				
Modification:	**Mar 06 2025 03:55 AM**				
Description:	A load balancing mechanism must be in place to distribute traffic evenly.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-08**				
Justify why meeting SP-08-02 can contribute to the fulfilment of UP-08	Prevents server overload, ensuring optimal system performance.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Performance Requirements

Table 4.74. System NonFunctional Requirements: SP-09-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-09-01**				
Creation:	**Mar 06 2025 03:57 AM**				
Modification:	**Mar 06 2025 03:59 AM**				
Description:	The system should be built with modular components for easy updates and new feature integration.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-09**				
Justify why meeting SP-09-01 can contribute to the fulfilment of UP-09	Simplifies addition of new features and bug fixes, allowing seamless expansion.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Performance Requirements

Table 4.75. System NonFunctional Requirements: SP-09-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-09-02**				
Creation:	**Mar 06 2025 03:59 AM**				
Modification:	**Mar 06 2025 04:00 AM**				
Description:	The system’s backend architecture should support horizontal scaling, allowing the addition of new servers to handle increasing loads.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-09**				
Justify why meeting SP-09-02 can contribute to the fulfilment of UP-09	Ensures long-term scalability and prevents system overhauls as usage increases.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Performance Requirements

Table 4.76. System NonFunctional Requirements: SP-09-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-09-03**				
Creation:	**Mar 06 2025 04:01 AM**				
Modification:	**Mar 06 2025 04:01 AM**				
Description:	Adding new schools or user accounts should require minimal configuration changes and should not affect existing users.				
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-09**				
Justify why meeting SP-09-03 can contribute to the fulfilment of UP-09	Ensures system efficiency even as the number of users grows.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Performance Requirements

Table 4.77. System NonFunctional Requirements: SP-01-01

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SP-01-01			Type / Functional / Non-Functional	
Creation:	Mar 04 2025 01:29 PM				
Modification:	Mar 06 2025 03:26 AM			User ☐ ☐	
				System ☐ ☒	
Description:	All data at rest must be encrypted using AES-256 encryption.			Product (sub-type below)	
				Availability/Reliability/Security	
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UP-01				
Justify why meeting SP-01-01 can contribute to the fulfilment of UP-01	Prevents unauthorized access to stored data, enhancing security.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	TC-001			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.78. System NonFunctional Requirements: SP-01-02

Project Name:	Elementary School Student Management Software Proposal				
Requirement #:	SP-01-02			Type / Functional / Non-Functional	
Creation:	Mar 06 2025 03:27 AM				
Modification:	Mar 06 2025 03:27 AM			User ☐ ☐	
				System ☐ ☒	
Description:	All data in transit must be encrypted using TLS 1.3 to protect against interception.			Product (sub-type below)	
				Availability/Reliability/Security	
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:	UP-01				
Justify why meeting SP-01-02 can contribute to the fulfilment of UP-01	Ensures secure communication and protects sensitive information.				
Traceability:	Use cases cf.	N/A			
	Test cases cf.	TC-001			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.79. System NonFunctional Requirements: SP-03-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-03-01**				
Creation:	**Mar 04 2025 01:34 PM**				
Modification:	**Mar 06 2025 03:29 AM**				
Description:	The system must restrict access to data based on user roles (e.g., administrator, teacher, parent).				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UP-03**				
Justify why meeting SP-03-01 can contribute to the fulfilment of UP-03	Ensures that sensitive information is only accessible to authorized users.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Availability/Reliability/Security

Table 4.80. System NonFunctional Requirements: SP-03-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-03-02**				
Creation:	**Mar 06 2025 03:29 AM**				
Modification:	**Mar 06 2025 10:50 AM**				
Description:	Multi-factor authentication (MFA) for admin-level accounts, using SMS or authentication apps.				
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Engineered From:	**UP-03**				
Justify why meeting SP-03-02 can contribute to the fulfilment of UP-03	Adds additional level of authenticated authorized user access				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Availability/Reliability/Security

Table 4.81. System NonFunctional Requirements: SP-04-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-04-01**				
Creation:	**Mar 06 2025 03:34 AM**				
Modification:	**Mar 06 2025 03:34 AM**				
Description:	The system must achieve 99.9% uptime, ensuring that planned maintenance occurs during non-peak hours.				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UP-04**				
Justify why meeting SP-04-01 can contribute to the fulfilment of UP-04	Minimizes disruptions for users who rely on the system for daily operations.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Availability/Reliability/Security

Table 4.82. System NonFunctional Requirements: SP-04-02

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-04-02**				
Creation:	**Mar 06 2025 03:34 AM**				
Modification:	**Mar 06 2025 03:35 AM**				
Description:	The system must perform automatic backups every 24 hours, with an option for administrators to trigger manual backups.				
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UP-04**				
Justify why meeting SP-04-02 can contribute to the fulfilment of UP-04	Prevents data loss and ensures data recovery in case of system failures.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Type	**Functional**	**Non-Functional**
User	☐	☐
System	☐	☒

Product (sub-type below)

Availability/Reliability/Security

Table 4.83. System NonFunctional Requirements: SP-04-03

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SP-04-03**			**Type**	**Functional** / **Non-Functional**
Creation:	**Mar 06 2025 03:35 AM**			**User**	☐ / ☐
Modification:	**Mar 06 2025 03:35 AM**			**System**	☐ / ☒
Description:	In case of failure, the system must restore data within 30 minutes from the latest backup.			**Product (sub-type below)**	
				Availability/Reliability/Security	
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:	**UP-04**				
Justify why meeting SP-04-03 can contribute to the fulfilment of UP-04	Reduces downtime and ensures business continuity.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.84. System NonFunctional Requirements: SE-01-01

Project Name:	**Elementary School Student Management Software Proposal**				
Requirement #:	**SE-01-01**			**Type**	**Functional** / **Non-Functional**
Creation:	**Mar 06 2025 10:45 AM**			**User**	☐ / ☐
Modification:	**Mar 06 2025 10:47 AM**			**System**	☐ / ☒
Description:	All personal data must be stored and handled according to FERPA, GDPR, and other relevant data privacy laws.			**External (sub-type below)**	
				Legislative Requirements on Safety/Security	
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:	**UE-01**				
Justify why meeting SE-01-01 can contribute to the fulfilment of UE-01	This ensures the system is meeting legal standards of information security and data protection.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.85. Mapping from user requirements to system requirements

Project Name: Elementary School Student Management Software Proposal			
User Requirements		**System Requirements**	
Req ID	**Description**	**Req ID**	**Description**
UE-01	System must securely store all personal information, sensitive or otherwise.	SE-01-01	All personal data must be stored and handled according to FERPA, GDPR, and other relevant data privacy laws.
UF-D	Users must be able to log into application and gain access to authorized information and functionalities based on their assigned user role (Admin, Teacher, Parent, Student).	SF-D-01	The system shall authenticate users using email/username and password before granting access.
		SF-D-02	The system shall implement role-based access control (RBAC) to restrict access based on user roles.
UF-E	Parents with multiple children should be able to select and access each child's information separately.	SF-E-01	The system shall allow a parent to be linked to multiple children.
		SF-E-02	The system shall provide a child selection dashboard for parents to switch between child profiles.
		SF-E-03	The system shall restrict access to only the children associated with the parent's account.
UF-F	Student Records will be used to store student details, enrollment history and academic progress, allowing admins to customize data and aesthetics relevant to school, teachers to maintain information and parents/students to view records.	SF-F-01	The system shall securely store student records, including personal details, enrollment history, and academic progress in a centralized database.
		SF-F-02	The system shall allow admins to define and customize data fields and aesthetics (e.g., layout or additional custom fields) via an administrative interface.
		SF-F-03	Teachers shall have write access to update student records, while parents and students shall have read-only access.
UF-H	Using a Course Management feature, teachers will be able to create, manage, and organize and parents/students will be able to view course materials of varying content and media types.	SF-H-01	The system shall allow teachers to create, manage, and organize course content (lessons, assignments, assessments, multimedia files) by subject, grade, and module.
		SF-H-02	The system shall support various content types including documents, videos, and interactive elements (e.g., quizzes).
		SF-H-03	The system shall allow teachers to set deadlines for assignments.
		SF-H-04	The system shall automatically display these materials in the student and parent portals.
		SF-I-01	The system shall allow teachers to record, edit, and manage grades for assignments, tests, and projects.

UF-I	A Gradebook will allow teachers to record and manage, and allow parents to view grades for assignments, tests, and projects with support for various grading scales.	SF-I-02	The system shall support multiple grading scales (letter, numeric, pass/fail).
		SF-I-03	The system shall calculate overall grades based on weighted assessments.
		SF-I-04	The system shall ensure that parents and students have read-only access to grade information.
UF-P	An Attendance Tracking feature will enable teachers to log attendance in real-time and automatically notify parents about absences or tardiness.	SF-P-01	The system shall automatically notify parents via email/SMS when a student is absent or tardy.
		SF-P-02	The system shall provide an interface (using checkboxes or dropdowns) for teachers to mark student attendance in real time.
UF-Q	A Parent-Teacher Communication feature must provide a two-way messaging system for personal communication and notifications for important updates.	SF-Q-01	The system shall provide a secure, two-way messaging platform for communication between parents and teachers, including support for text messages and file attachments.
		SF-Q-02	The system shall notify users (via email or SMS) when a new message is received
UF-R	An Announcement feature should enable admins or teachers to post group specific custom announcements to be displayed in a dashboard.	SF-R-01	The system shall allow admins/teachers to create announcements.
		SF-R-02	The system shall allow announcements to be targeted by grade, class, or school-wide.
		SF-R-03	Announcements shall be displayed on the user dashboard.
UF-S	Parents should be able to subscribe and unsubscribe to selected Alert Notifications via email or SMS.	SF-S-01	The system shall allow parents to enable or disable specific notifications (e.g., attendance, grades, announcements) through a user preference interface.
		SF-S-02	The system shall send alerts via email and SMS based on user preferences.
UP-01	All data must be encrypted during storage and transmission.	SP-01-01	All data at rest must be encrypted using AES-256 encryption.
		SP-01-02	All data in transit must be encrypted using TLS 1.3 to protect against interception.
UP-02	Comprehensive user support should be available.	SP-02-01	The system must provide user documentation including tooltips, FAQs, and step-by-step guides available from an in-app help center.
		SP-02-02	System must provide an interactive website tutorial or inline guidance for key features to assist first-time users.
		SP-02-03	System should be designed with a modular architecture
UP-03	Role-Based Access Control (RBAC) must ensure users access only authorized data.	SP-03-01	The system must restrict access to data based on user roles (e.g., administrator, teacher, parent).
		SP-03-02	Multi-factor authentication (MFA) for admin-level accounts, using SMS or authentication

			apps.
UP-04	The system must maintain 99.9% uptime, with automatic backups and a fast recovery time in case of failure.	SP-04-01	The system must achieve 99.9% uptime, ensuring that planned maintenance occurs during non-peak hours.
		SP-04-02	The system must perform automatic backups every 24 hours, with an option for administrators to trigger manual backups.
		SP-04-03	In case of failure, the system must restore data within 30 minutes from the latest backup.
UP-05	The system must have an intuitive, user-friendly interface whose commonly used features are easily accessed.	SP-05-01	The system must have a clean, consistent design with a clear layout to facilitate navigation.
		SP-05-02	Frequently accessed features should be accessible with one click from a menu anchored to every page.
		SP-05-03	The system must maintain a consistent and responsive UI across all pages with a standardized design language.
		SP-05-04	Icons, labels, and tooltips must be used effectively to reduce reliance on text-heavy navigation.
UP-06	Real-time updates (e.g., attendance, notifications) should occur with minimal delay.	SP-06-01	The system must use WebSockets or push notifications to enable real-time updates.
		SP-06-02	The notification system must allow asynchronous processing to prevent bottlenecks.
		SP-06-03	Attendance logs, grade updates, and notifications must be processed within 2 seconds.
UP-07	The system must be compatible with modern web browsers and mobile devices, ensuring a responsive design for smartphones and tablets.	SP-07-01	The system must be compatible with Chrome, Firefox, Safari, and Edge, and should maintain usability across desktop and mobile devices.
		SP-07-02	The system must be mobile-responsive, automatically adjusting UI elements based on screen size.
		SP-07-03	All features must be accessible on both desktop (≥1024px screen width) and mobile (≥5 inches screen width) without requiring a separate app.
UP-08	The system must handle up to 1000 concurrent users without performance degradation.	SP-08-01	The system must support 1000 concurrent users while maintaining response times below 3 seconds.
		SP-08-02	A load balancing mechanism must be in place to distribute traffic evenly.
		SP-09-01	The system should be built with modular components for easy updates and new feature integration.

UP-09	The system must be scalable, supporting additional schools, grades, and users without significant reconfiguration.	SP-09-02	The system’s backend architecture should support horizontal scaling, allowing the addition of new servers to handle increasing loads.
		SP-09-03	Adding new schools or user accounts should require minimal configuration changes and should not affect existing users.

Table 8.2.1. Test Suite TS-001: Secure Data Storage and Encryption

Test Case ID	Test Stage	Test Case Description	Tested
TC-001	System	Verify personal/sensitive data is encrypted at rest (AES-256) and in transit (TLS 1.3).	Yes

Table 8.2.2. Test Suite TS-002: Role-Based Login and Access

Test Case ID	Test Stage	Test Case Description	Tested
TC-002	Integration	Ensure users log in with credentials and access only authorized features.	Yes

Table 8.2.3. Test Suite TS-003: Parent Multi-Child Profile Access

Test Case ID	Test Stage	Test Case Description	Tested
TC-003	System	Parent with 2+ children can switch profiles and access each child’s data.	Yes

Table 8.2.4. Test Suite TS-004: Student Record Customization by Admin

Test Case ID	Test Stage	Test Case Description	Tested
TC-004	System	Admin adds a student to student records.	Yes

Table 8.2.5. Test Suite TS-005: Two-Way Parent-Teacher Messaging

Test Case ID	Test Stage	Test Case Description	Tested
TC-005	System	Parent sends a message to Teacher;	Yes

Table 8.2.6. Test Suite TS-006: Real-Time Attendance Alerts

Test Case ID	Test Stage	Test Case Description	Tested
TC-006	System	Teacher marks a student as absent; parent receives SMS/email within 2 seconds.	Yes

Table 8.2.7. Test Suite TS-007: Gradebook Calculations and Access

Test Case ID	Test Stage	Test Case Description	Tested
TC-007	Integration	Teacher inputs grades with a weighted scale; parent views read-only results.	Yes

Table 8.2.8. Test Suite TS-008: Targeted Announcements

Test Case ID	Test Stage	Test Case Description	Tested
TC-008	System	Admin posts an announcement for “Grade 5 Only.”	Yes

Table 8.2.9. Test Suite TS-009: Notification Subscription Preferences

Test Case ID	Test Stage	Test Case Description	Tested
TC-009	System	Parent unsubscribes from grade alerts but stays subscribed to attendance.	Yes

Table 8.2.10. Test Case TC-002

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-002: Role-Based Login and Access	
Test Case ID	TC-002 (Integration Test)	
What To Test	Ensure users log in with credentials and access only authorized features.	
Test Data Input	Login using Admin Credentials, Login Using Parent Credentials	
Expected Result	Admin sees dashboard with user management tools. Parent sees only child records and grades.	
Traceability	**Relevant User Req.(s)**	UF-D
	Relevant System Req.(s)	SF-D-01,SF-D-02
	Relevant Use Case(s)	UC-001
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.11. Test Case TC-007

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-007: Gradebook Calculations and Access	
Test Case ID	TC-007 (Integration Test)	
What To Test	Teacher inputs grades with a weighted scale; parent views read-only results.	
Test Data Input	Teacher enters grades.	
Expected Result	System calculates overall grade. Parent sees grades but cannot edit them.	
Traceability	**Relevant User Req.(s)**	UF-I
	Relevant System Req.(s)	SF-I-01,SF-I-02,SF-I-03,SF-I-04
	Relevant Use Case(s)	UC-007,UC-008
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.12. Test Case TC-001

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-001: Secure Data Storage and Encryption	
Test Case ID	TC-001 (System Test)	
What To Test	Verify personal/sensitive data is encrypted at rest (AES-256) and in transit (TLS 1.3).	
Test Data Input	Submit a student record via the UI.	
Expected Result	Data is encrypted in the database	
Traceability	**Relevant User Req.(s)**	UE-01,UP-01
	Relevant System Req.(s)	SP-01-01,SP-01-02
	Relevant Use Case(s)	
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.13. Test Case TC-003

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-003: Parent Multi-Child Profile Access	
Test Case ID	TC-003 (System Test)	
What To Test	Parent with 2+ children can switch profiles and access each child’s data.	
Test Data Input	Parent logs in and selects Child A, then Child B from a dropdown.	
Expected Result	Child A’s grades/attendance display first. Switching to Child B updates the UI to Child B’s data.	
Traceability	**Relevant User Req.(s)**	UF-E
	Relevant System Req.(s)	SF-E-01,SF-E-02,SF-E-03
	Relevant Use Case(s)	UC-003
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.14. Test Case TC-004

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-004: Student Record Customization by Admin	
Test Case ID	TC-004 (System Test)	
What To Test	Admin adds a student to student records.	
Test Data Input	Admin navigates to Settings > Student Records > adds the field.	
Expected Result	Teachers can input data into the new field.	
Traceability	**Relevant User Req.(s)**	UF-F
	Relevant System Req.(s)	SF-F-01,SF-F-02
	Relevant Use Case(s)	
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.15. Test Case TC-005

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-005: Two-Way Parent-Teacher Messaging	
Test Case ID	TC-005 (System Test)	
What To Test	Parent sends a message to Teacher;	
Test Data Input	Parent sends “Is the field trip confirmed?	
Expected Result	Both users see the message thread. Parent receives SMS/email notification for the reply.	
Traceability	**Relevant User Req.(s)**	UF-Q
	Relevant System Req.(s)	SF-Q-01,SF-Q-02
	Relevant Use Case(s)	UC-005
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.16. Test Case TC-006

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-006: Real-Time Attendance Alerts	
Test Case ID	TC-006 (System Test)	
What To Test	Teacher marks a student as absent; parent receives SMS/email within 2 seconds.	
Test Data Input	Teacher selects “Absent” for Student X in the attendance UI.	
Expected Result	Parent’s phone/email receives notification. Logs show alert timestamp ≤2 seconds after submission.	
Traceability	**Relevant User Req.(s)**	UF-P
	Relevant System Req.(s)	SF-P-01,SF-P-02,SP-06-03
	Relevant Use Case(s)	UC-006
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.17. Test Case TC-008

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-008: Targeted Announcements	
Test Case ID	TC-008 (System Test)	
What To Test	Admin posts an announcement for “Grade 5 Only.”	
Test Data Input	Admin creates announcement with target = Grade 5.	
Expected Result	Grade 5 parents/students see the announcement while other grades do not.	
Traceability	**Relevant User Req.(s)**	UF-R
	Relevant System Req.(s)	SF-R-01,SF-R-02,SF-R-03
	Relevant Use Case(s)	UC-004
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.2.18. Test Case TC-009

Project Name:	**Elementary School Student Management Software Proposal**	
Test Suite	TS-009: Notification Subscription Preferences	
Test Case ID	TC-009 (System Test)	
What To Test	Parent unsubscribes from grade alerts but stays subscribed to attendance.	
Test Data Input	Parent toggles off “Grade Alerts” in Settings > Notifications.	
Expected Result	Parent receives attendance alerts but no grade updates.	
Traceability	**Relevant User Req.(s)**	UF-S
	Relevant System Req.(s)	SF-S-01,SF-S-02
	Relevant Use Case(s)	UC-002
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 8.3.1. Execution Report of Test Case TC-002

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-002	
Testing Tools Used:	None	
Testing Type:	Component interface testing	
Execution Steps:	1	Log in as Admin
	2	Verify access to user management tools.
	3	Log in as Parent
	4	Confirm Parent cannot access admin features.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Parent Accessed Admin Management tools	Fail	Parent had has admin access	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	Parent Cannot Access Admin features	Pass	None	05/08/2025 by Leonardo Cruz

Execution Summary:	Success

Table 8.3.2. Execution Report of Test Case TC-007

Project Name:	**Elementary School Student Management Software Proposal**
Test Case ID:	TC-007
Testing Tools Used:	None
Testing Type:	Component interface testing
Execution Steps:	1 Teacher inputs grades with weights
	2 Verify system calculates overall grade correctly.
	3 Parent views read-only grades.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Grade was successfully saved	Fail	Grade is not calculated using weighted percentiles	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	Grades are entered and calculated accordingly for parents to see	Pass	None	05/08/2025 by Leonardo Cruz

Execution Summary:	Success

Table 8.3.3. Execution Report of Test Case TC-001

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-001	
Testing Tools Used:	Wireshark	
Testing Type:	Security testing	
Execution Steps:	1	Log in as an admin and navigate to the student record
	2	Enter a test student’s personal data
	3	Submit the record and capture network traffic using Wireshark.
	4	Verify encryption status for the stored data and TLS version in network traffic.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Data stored in plaintext; TLS 1.2 detected.	Fail	Failed due to missing encryption configuration.	04/24/2025 by Leonardo Cruz

Execution Summary:	Failed

Table 8.3.4. Execution Report of Test Case TC-003

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-003	
Testing Tools Used:	None	
Testing Type:	Functional testing	
Execution Steps:	1	Parent logs in and selects Child A from the dropdown.
	2	Verify Child A's attendance/grades display.
	3	Switch to Child B's profile.
	4	Confirm Child B's data replaces Child A's.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Parental access to all students	Fail	Parental access to all students	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	Parents can see the data for their students	Pass	None	

Execution Summary:	Success

Table 8.3.5. Execution Report of Test Case TC-004

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-004	
Testing Tools Used:	None	
Testing Type:	Functional testing	
Execution Steps:	1	Admin accesses student record
	2	Admin adds a student
	3	Verify the student appears in student records.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Student has been successfully added	Pass	None	04/25/2025 by Leonardo Cruz

Execution Summary:	Success

Table 8.3.6. Execution Report of Test Case TC-005

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-005	
Testing Tools Used:	None	
Testing Type:	Functional testing	
Execution Steps:	1	Parent sends a message to Teacher.
	2	Teacher receives message in the inbox
	3	Verify both users receive notifications.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/25/2025	Teacher not receiving message	Fail	No Database is connected to inbox	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	Teachers and Parents can both send and recieve messages to each other	Pass	None	05/08/2025 by Leonardo Cruz

Execution Summary:	Success

Table 8.3.7. Execution Report of Test Case TC-006

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-006	
Testing Tools Used:	None	
Testing Type:	Functional testing	
Execution Steps:	1	Teacher accesses the Attendance
	2	Teacher marks a student as "Absent."
	3	Monitor SMS/email to parent.
	4	Verify alert timestamp within 2 seconds.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Student enters into database, But no notification	Fail	No Notification system	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	Parent Recieves email notification with absent notice	Pass	None	05/08/2025 by Leonardo Cruz

Execution Summary:	Success

Table 8.3.8. Execution Report of Test Case TC-008

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-008	
Testing Tools Used:	None	
Testing Type:	Functional testing	
Execution Steps:	1	Admin accesses the Announcements page
	2	Admin posts announcement for "Grade 1 Only."
	3	Log in as Grade 1 Parent and verify visibility.
	4	Log in as Grade 4 Parent; confirm no announcement.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	No function for filtering announcement targets	Fail	No function for filtering announcement targets	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	Admin post announcements only 1st graders and their parents can see	Pass	None	05/08/2025 by Leonardo Cruz

Execution Summary:	Success

Table 8.3.9. Execution Report of Test Case TC-009

Project Name:	**Elementary School Student Management Software Proposal**	
Test Case ID:	TC-009	
Testing Tools Used:	None	
Testing Type:	Usability testing	
Execution Steps:	1	Parent unsubscribes from "Grade Alerts."
	2	Teacher posts a grade; verify no alert sent.
	3	Teacher logs attendance; verify alert sent.

Test Execution Records:

#	Tester	Test Date	Actual Result	Status	Defect	Correction
1	Leonardo Cruz	04/24/2025	Grade post sucessfully, No notifications.	Fail	No notification has been set up yet	04/24/2025 by Leonardo Cruz
2	Leonardo Cruz	05/08/2025	No option to subscribe for grades yet, but alerts for attendance still functions	Fail	No subscription options for parent.	05/08/2025 by Leonardo Cruz

Execution Summary:	Failed