SOFTWARE REQUIREMENTS SPECIFICATION

for

JDBC School Management Project

Version 1.0

Prepared by : 1. Rutul Patel (IMT2022021)
2. Varnit Mittal (IMT2022025)
3. Aditya Priyadarshi (IMT2022075)
4. Hemang Seth (IMT2022098)

Submitted to : Sujit Kumar Chakrabarti
Lecturer

November 6, 2024

Contents

1 Introduction

1.1 Purpose

The purpose of this **Software Requirements Specification (SRS)** document [**?**]is to define the comprehensive requirements for the `JDBC`-based School Management Project. This project aims to develop a **robust**, **efficient**, and **scalable** school management system that leverages *Java Database Connectivity (JDBC)* to seamlessly integrate with a relational database. The proposed system will streamline administrative tasks, enhance data integrity, and improve the overall management of school operations, including **student information**, **staff management**, **class schedules**, and the **library management system** for the school. By defining clear *functional* and *non-functional* requirements, this document serves as a foundation for development, ensuring that the final system aligns with the school's operational needs and enhances the **user experience** for both administrators and faculty.

1.2 Document Conventions

This SRS follows IEEE standards for structured documentation, using consistent typographical conventions. Headings are in bold, and requirements are numbered sequentially for easy reference. High-level requirement priorities are inherited by detailed requirements, with each major requirement statement retaining its own priority level. Important in-between terms are also in bold.

1.3 Intended Audience and Reading Suggestions

This SRS document is intended for all stakeholders involved in the JDBC School Management Project, including:

- **Project Managers:** To understand the scope and objectives for effective planning.
- **Developers and Database Administrators:** To review system requirements and database configurations.
- **Quality Assurance Teams:** To design test cases based on outlined functionalities.
- **School Administrators and Users:** To understand system features for training and usage.

1.4 Project Scope

The JDBC School Management Project aims to develop a comprehensive software system for streamlining school operations, including student information management, staff records, scheduling, and library management system. Leveraging Java Database Connectivity (JDBC), the project will integrate a user-friendly Java application with a relational database, enabling secure, real-time access to data. The system will address the limitations of manual record-keeping by automating core administrative tasks, reducing errors, and ensuring data consistency. Key stakeholders—school administrators, teachers, and support staff—will benefit from the system's efficiency, as it will facilitate smoother communication, data accessibility, and decision-making.

The scope of the project includes the development of user roles for different levels of access, including administrators, teachers, and support staff, each tailored to their responsibilities. Administrators will have comprehensive access to manage all school operations, while teachers will focus on student performance. Role-based access ensures data security, preventing unauthorized access and maintaining user-specific functionalities.

Key components of the project include:

- **Data Integrity and Consistency:** Built-in database constraints and validation checks will ensure data accuracy, preventing duplication and maintaining up-to-date records.
- **Real-Time Reporting:** Customizable reporting tools for generating real-time insights on student performance, faculty workload, financials, and overall school metrics.
- **Role-Based Security:** A secure authentication and authorization framework to provide role-specific access and maintain sensitive data privacy.
- **Adaptability for Future Growth:** The system is designed with scalability in mind, allowing for future upgrades, additional modules, or integration with other educational tools or databases.

Figure 1.1 (Entire work-flow) is the overview of the project.

- **Student Management:** Handles the addition, update, removal, and retrieval of student details.
- **Course Management:** Facilitates course creation, modification, deletion, and listing of courses by teacher.
- **Library Management:** Manages books, links them to courses, and allows removal and viewing of course materials.
- **Teacher Management:** Manages teacher records, including addition, update, removal, and retrieval.

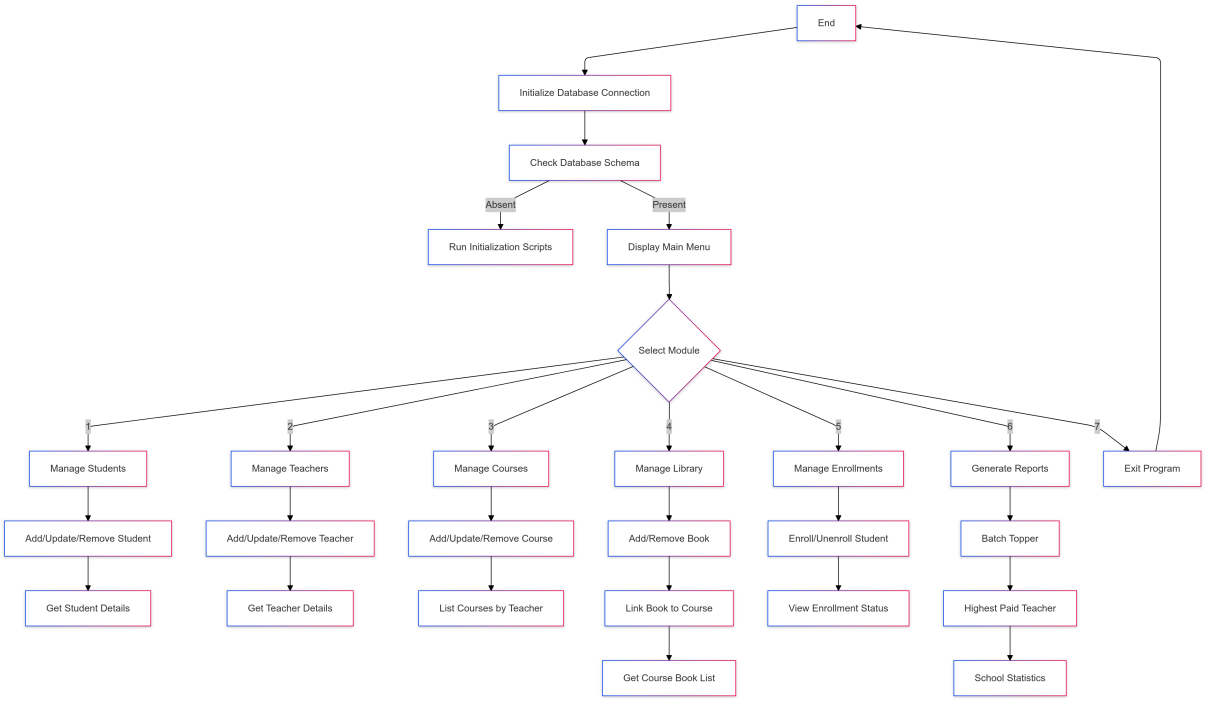

Figure 1.1: Entire work-flow

1.5 References

1. IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications," IEEE, 1998.

2. R. Pressman, *Software Engineering: A Practitioner's Approach*, 9th ed. McGraw-Hill, 2014.

2 Overall Description

2.1 Product Perspective

The proposed system is an integrated solution for managing academic and administrative operations in educational institutions. It aims to streamline processes such as student, teacher, course, enrollment, and library management, thereby reducing manual tasks and improving data accuracy. The system will serve administrators, faculty, and students, providing efficient tools for data management, reporting, and course material handling. It will offer a secure, scalable web-based platform with a user-friendly interface, capable of integrating with existing institutional databases. By providing real-time data access and analytics, the system will support informed decision-making and enhance operational efficiency.

2.2 User Classes and Characteristics

"JDBC School Management Tool" has basically 4 types of users.

- Teachers
- Students
- Admin
- Librarian

The Teacher manages course content and interacts with students, while the Student accesses materials, tracks progress, and enrolls in courses. The Admin oversees the system, manages users, and generates reports, and the Librarian manages library resources and book loans.

2.3 Product Functions

The **Use Case Diagram** for the School Management System provides an overview of the interactions between different actors and the system. It defines the primary functionalities available to each user role, along with the system's core operations related to student, teacher, course, and library management. The system includes the following key actors and their respective interactions:

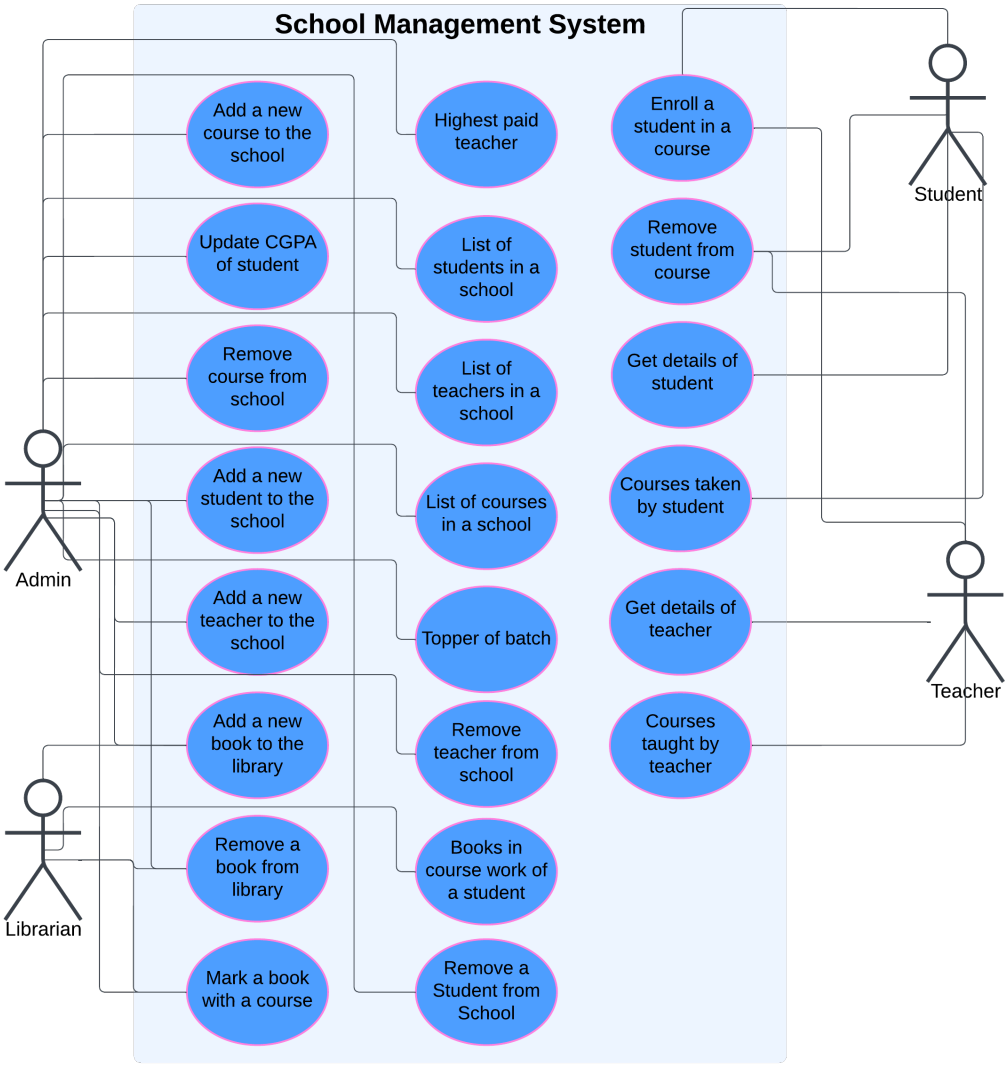

Figure 2.1: Type of Users

2.3.1 Actors

- **Admin:** Responsible for high-level management tasks, including adding and removing students, teachers, and courses. The admin also updates critical data such as student CGPA and teacher salaries, retrieves lists of all students, teachers, and courses in the school, and determines the highest-achieving student and the highest-paid teacher.
- **Student:** Enrolls in and removes themselves from courses, checks their details, and views the list of courses in which they are enrolled.
- **Teacher:** Manages student enrollments and removals within courses, checks their personal details, and reviews the courses they teach.
- **Librarian:** Manages the library catalog by adding and removing books, associating books with specific courses, and reviewing books relevant to students' coursework.

2.3.2 Core Use Cases:

- **Student Management:** Includes functionalities to add or remove students, view student details, and update a student's CGPA. Admin handles student additions/removals and CGPA updates, while students can enroll in or withdraw from courses.
- **Teacher Management:** Involves adding and removing teachers, adjusting teacher salaries, and obtaining teacher details. Teachers can enroll or remove students from courses, view their own information, and see the courses they teach.
- **Course Management:** The admin can add or remove courses, view all available courses, and link books to courses. Both students and teachers can interact with courses based on enrollment.
- **Library Management:** The admin and librarian handle adding/removing books and associating books with courses. Librarians also retrieve information about books in students' course materials.

Each use case is designed to facilitate seamless communication between actors and system functionalities, supporting efficient school administration and resource management.

This **Data Flow Diagram (DFD)** illustrates the high-level architecture of a school management system, outlining key entities, processes, and data stores for an SRS document.

2.3.3 Entities

- **Admin**: Central authority responsible for overseeing processes related to student, teacher, and course management.
- **Teacher, Student, Librarian:** System users interacting with various modules—teachers manage grades and courses, students handle enrollment and grade requests, and librarians oversee library inventory.

2.3.4 Processes

- **Information Management Modules:** Separate modules for managing Student, Teacher, and Course information, processing requests and maintaining their respective records.
- **Enrollment, Grade, and Library Management:** Processes focused on student enrollment, grade submission, and library inventory, interfacing with relevant databases.

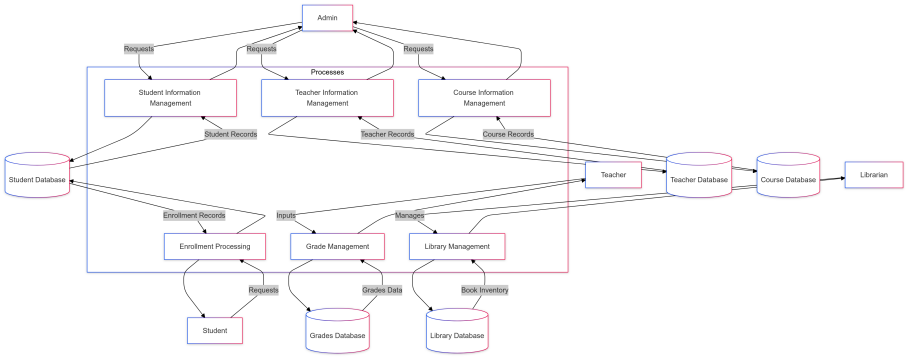

Figure 2.2: Data Flow Diagram

2.3.5 Data Stores

- **Student, Teacher, and Course Databases:** Hold core records for students, teachers, and courses, updated by respective processes.
- **Grades and Library Databases:** Specialized storage for grades and library inventory, supporting academic and resource management.

2.3.6 Data Flow

Data flows between entities, processes, and databases, with arrows indicating updates, requests, and record exchanges. This DFD provides an overview of system interactions and data dependencies at a high level, suitable for specifying functional requirements in the SRS.

2.4 Operating Environment

The website will be operate in any Operating Environment - Mac, Windows, Linux etc.

2.5 Design

- **School Class:** Represents the school itself. It aggregates Students, Teachers, and Courses (indicating that a school can manage multiple students, teachers, and courses). It has methods for adding and removing these entities.
- **Person Class:** This is a base class inherited by Student and Teacher. It contains common properties like name, date of birth, and address.
- **Student Class:** Inherits from Person and has attributes specific to students (e.g., roll number, CGPA). Students can enroll in courses and have their CGPA updated.
- **Teacher Class:** Also inherits from Person and has attributes specific to teachers (e.g., employee ID, salary). Teachers can teach courses and have their salary incremented.
- **Course Class:** Represents a course and can contain a list of books. Courses can be taken by students and taught by teachers.
- **Book Class:** Represents books in the library and can be marked with specific courses.
- **Library Class:** Represents the school's library and holds books.
- **Associations:**
  - **Aggregation:** A school aggregates students, teachers, and courses. This means that the school can exist independently of the students, teachers, and courses, but these entities are part of the school.
  - **Composition:** A course is composed of a teacher, meaning that a course cannot exist without a teacher.

  Students are enrolled in courses, teachers teach courses, and courses can have books.
- **Inheritance:** Both Student and Teacher inherit from Person, sharing common attributes.

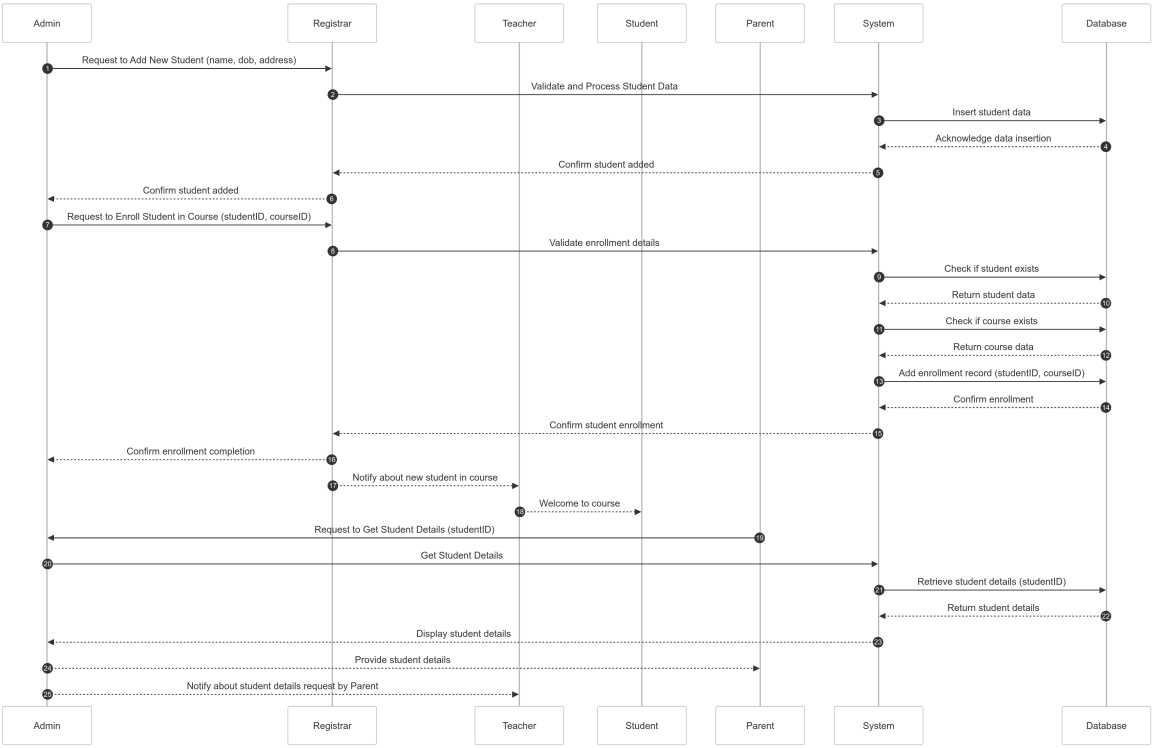

Figure 2.3: Student Activities

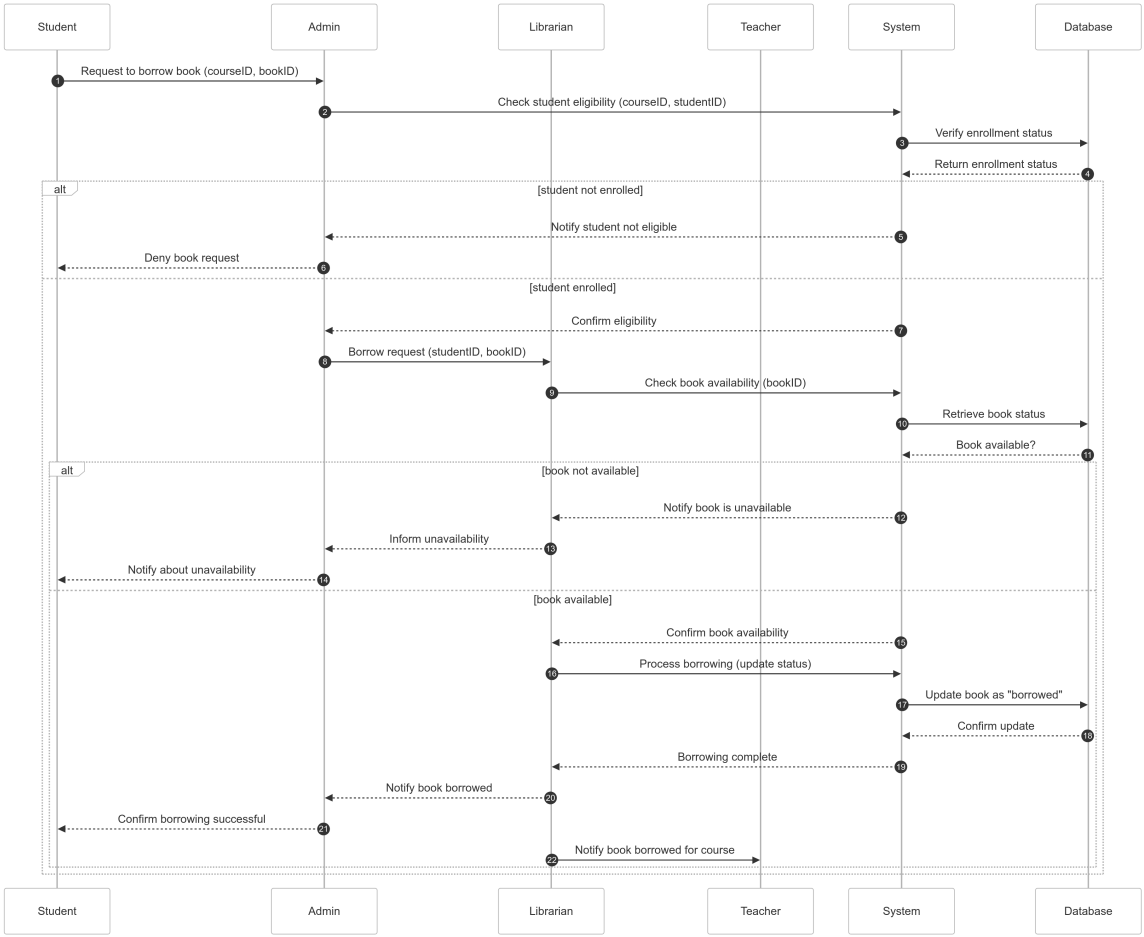

Figure 2.4: Teacher Activities

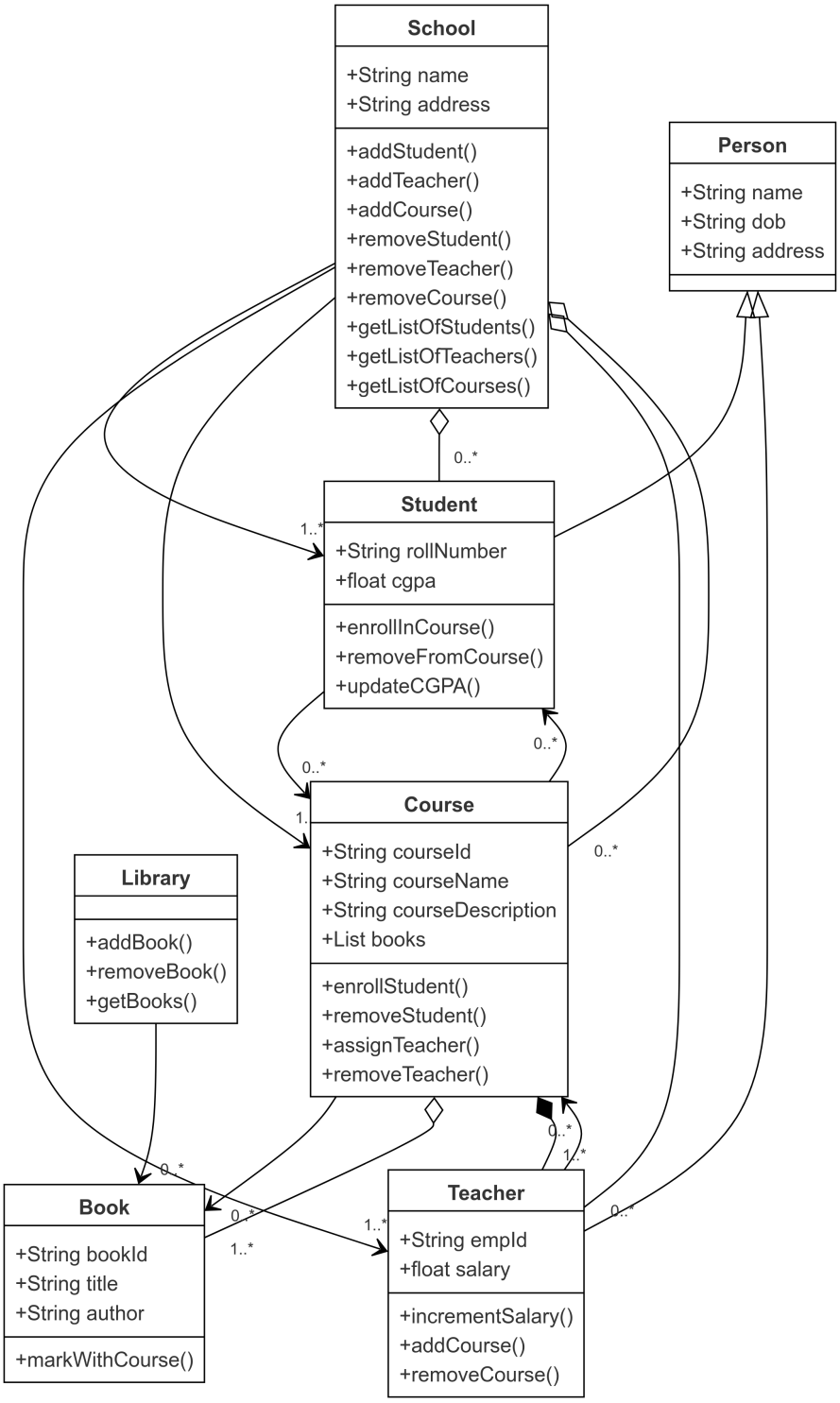

Figure 2.5: Class Diagram for JDBC School Management Tool

3 System Features

The JDBC-based School Management System streamlines the management of students, teachers, courses, and library resources. Key features include student enrollment, course management, grade entry, and CGPA calculation. It also allows administrators to manage teacher and student records, update information, and generate reports. Designed for efficiency and ease of use, the system ensures data integrity, security, and accessibility.

3.1 Description and Priority

The JDBC-based School Management System encompasses core and supporting features essential to streamline school operations. Each feature has been prioritized based on its impact on functionality, user experience, and overall system objectives. The list below ranks features in descending order of importance, highlighting their roles and intended users.

1. **Student Enrollment and Management**: This foundational feature allows administrators to add, update, or remove students from the school database. It supports core data management functions, ensuring that student records are current and accessible. Given its importance for school record-keeping, this feature is essential and operated by administrators and authorized personnel.

2. **Course Assignment and Management**: Designed to enable administrators to add or update course offerings, this feature allows assigning students and teachers to specific courses. It ensures the smooth allocation of resources and is crucial for planning and tracking academic programs. Managed primarily by administrative staff, this feature is high-priority to maintain organized course records.

3. **Teacher Assignment and Salary Management**: This feature empowers administrators to add or update teacher details, assign teaching roles, and manage salary records. It ensures accurate tracking of teacher assignments and allows adjustments to compensation when needed. This functionality is integral to human resource management within the school.

4. **Grades and CGPA Calculation**: Teachers use this feature to record student performance and generate cumulative scores, which are then aggregated to calculate CGPA. This process maintains academic integrity by offering a transparent grading system. Since accurate grade entry and GPA calculation are critical, this feature is high-priority for student evaluations.

5. **Library Resources and Course Material Management**: The system allows administrators to manage library resources, link books to specific courses, and maintain an updated list of available materials. This feature is particularly useful for both students and teachers to support their educational needs. It ranks high due to its importance in providing academic resources.

6. **Reports and Analytics**: Designed to generate detailed reports on student performance, teacher assignments, and overall academic trends, this feature is valuable for school directors and decision-makers. It provides insights into educational outcomes, enabling data-driven decision-making and continuous improvement. Though not immediately required, it holds substantial strategic value.

7. **User Access and Security Controls**: This feature enforces role-based access for users, limiting functionalities based on the user's role (e.g., administrator, teacher, or student). It ensures data security and compliance, protecting sensitive student and staff information. This feature is prioritized to maintain secure access and safeguard system integrity.

Each feature plays a significant role in achieving the system's objectives, enabling efficient management of school operations, resources, and academic records while ensuring secure access and data protection.

3.2 Functional Requirements

The JDBC-based School Management System offers streamlined functionalities across various operational areas. The functional requirements are structured into categories, focusing on essential management aspects for ease of navigation and understanding.

- **Student Management**
  - **Student Enrollment and Records**
    - **FR1**: Administrators shall be able to add new students to the system.
    - **FR2**: Administrators shall have the option to remove students from the system.
    - **FR3**: The system shall allow updates to a student's CGPA by authorized personnel.
  - **Information Access and Course Enrollment**
    - **FR4**: Administrators shall access detailed information for each student.
    - **FR5**: A list of enrolled courses per student shall be available for viewing.
    - **FR6**: The system shall generate a master list of all students.
- **Teacher Management**
  - **Teacher Enrollment and Payroll**

    - **FR7**: The system shall enable the addition of new teachers.
    - **FR8**: The system shall support the removal of teacher profiles.
    - **FR9**: Salary updates for teachers, including increments, shall be manageable.
  - **Access to Teacher Information and Roles**
    - **FR10**: Administrators shall retrieve information for each teacher.
    - **FR11**: The highest-paid teacher shall be identifiable within the system.
    - **FR12**: Administrators shall view lists of courses each teacher is teaching.

- **Course Management**
  - **Course Catalog and Availability**
    - **FR13**: The system shall support adding new courses.
    - **FR14**: Removal of courses from the catalog shall be possible.
    - **FR15**: Detailed information for each course shall be accessible.
  - **Performance Tracking**
    - **FR16**: The system shall identify the top performer in each batch.
    - **FR17**: A list of all active courses shall be generated on request.

- **Enrollment Management**
  - **FR18**: Administrators shall enroll and remove students in courses through the system when necessary.

- **Library Management**
  - **Library Inventory**
    - **FR19**: The system shall support adding new books to the library database.
    - **FR20**: Books shall be associated with relevant courses for student reference.
  - **Library Access and Management**
    - **FR21**: A list of books in each student's coursework shall be available.
    - **FR22**: Administrators shall remove books from the library as needed.

- **System Management**
  - **FR23**: The system shall provide users with a secure exit option.

4 Non-Functional Requirements

This chapter details the non-functional requirements essential for ensuring the JDBC-based School Management System meets the desired standards for quality, security, performance, and user satisfaction. Each section addresses critical system attributes, which collectively define the expected operation and robustness of the system.

4.1 Quality Attributes

- **Reliability**: The system should maintain high reliability to ensure continuous and error-free operations, particularly in scenarios involving multiple user interactions and database transactions. Data integrity must be preserved across all modules.
- **Scalability and Elasticity**: The system should be designed to handle an increasing number of students, teachers, and courses without a decrease in performance. The architecture must be adaptable to future expansions with minimal reconfiguration.
- **Modularity**: Each component, including student, teacher, course, and library management, should be modular, allowing independent development and testing. This modular design facilitates future feature enhancements or updates.
- **Availability**: The system should have an uptime of at least 99.9%, ensuring high availability for school administrators and faculty. Database connections via JDBC should be resilient to network interruptions.
- **Adaptability and Flexibility**: The system should support adaptability to allow integration with new educational policies or updated school management requirements. Flexibility in configurations should be incorporated for easy customization.

4.2 Performance Requirements

- **Response Time**: Key operations, such as student enrollment or course assignment, should have a response time of less than 2 seconds to ensure a smooth user experience.
- **Resource Efficiency**: The system should be optimized to consume minimal server memory and CPU resources. JDBC database connections must be managed efficiently to prevent resource leaks.

- **Concurrency**: The system should support concurrent access by multiple administrators and users without performance degradation. The use of transaction management within JDBC is crucial for handling concurrent database modifications.

4.3 Security Requirements

- **Data Protection**: Sensitive information such as student details, teacher salaries, and course data must be securely stored and protected from unauthorized access.
- **Access Control**: Role-based access should be enforced to ensure that only authorized personnel can perform specific actions. Administrators should have distinct permissions from general users.
- **Compliance with Data Privacy Regulations**: The system must adhere to data protection standards applicable to educational institutions, including data retention and secure deletion of sensitive data.

4.4 Usability and Accessibility

- **Ease of Use**: The interface should be intuitive for school staff, requiring minimal training. User-friendly error messages and tooltips should guide users through complex tasks.
- **Accessibility**: The system should be accessible to users with disabilities, following accessibility guidelines to accommodate a diverse user base.

4.5 Resource Usage

- **Processor and Memory Usage**: The system should be optimized to minimize memory and CPU usage, even with extensive data and multiple concurrent users.
- **Human Resources**: The system should be straightforward to maintain and upgrade, reducing the need for extensive technical support and allowing administrators to manage most tasks without specialized training.
- **Power Efficiency**: For energy efficiency, the server hosting the JDBC-based system should be optimized to prevent excess power usage, supporting sustainability goals.

4.6 Sustainability and Compliance

- **Sustainability Goals**: By digitalizing school records and automating processes, the system aligns with sustainable practices by reducing the need for paper-based record-keeping, contributing to environmental sustainability.

- **Regulatory Compliance**: The system must comply with educational data management standards, ensuring adherence to regulations on student and staff information security.

4.7 Software Quality Assurance

- **Testing and Validation**: Comprehensive testing, including unit, integration, and user acceptance testing, must be conducted to validate the functionality and reliability of each component.
- **Dependability and Stability**: Continuous integration practices should be applied to maintain system stability during updates or enhancements, ensuring the system's dependable operation.

5 Other Requirements

5.1 System Maintenance

The JDBC-based School Management System will require periodic maintenance to ensure its ongoing functionality. Updates and bug fixes will be necessary to adapt to evolving user needs and technological changes.

5.2 Performance and Scalability

The system should be designed to handle growing data and user traffic. Performance optimization and scalability are essential to maintain efficient operation, even with increased usage.

5.3 Flexibility and Adaptability

As requirements change, the system must be flexible enough to accommodate future features and modifications. It should allow for easy updates and integration with new modules as needed.