Project Final Report

Primary School Management System

NAME: Sakib Shahriar Rafi

Contents

1. Introduction

In the evolving landscape of educational institutions, efficient management systems play a pivotal role in streamlining administrative and academic processes. The proposed project, titled "Primary School Management System," aims to address this need by providing an integrated platform for the management of student, teacher, and staff information, course enrollment, grading, and financial transactions. The system has been designed to enhance the overall efficiency of a primary school's operations while ensuring data security and accuracy.

1.1. Proposed Goals

The initial project proposal outlined a comprehensive set of functionalities catering to various user roles within the school ecosystem, including administrators, students, teachers, and staff members. The proposal included features such as user authentication, user role-based access control, student enrollment management, course allocation, grading, result tracking, payment processing, and reporting. The proposal formed the foundation for our project's scope and objectives.

1.2. Project Accomplishment

Since the project's inception, significant progress has been made in materializing the proposed functionalities. We have successfully implemented a user authentication system, allowing different user roles to access the system securely. The graphical user interface (GUI) has been developed using Java's JFrame library, providing an intuitive platform for users to interact with the system.

Throughout the development process, we recognized the need to adapt certain aspects of the original proposal to ensure a more efficient and robust implementation. Specifically, the database table structure underwent modifications to enhance data consistency, optimize query performance, and accommodate future scalability. These adaptations were based on insights gained during the development process and feedback received from potential users.

The project's core objectives remain aligned with the initial proposal, aiming to create a system that empowers administrators, students, teachers, and staff members with tools to efficiently manage their respective tasks within the school environment.

In the following sections of this report, we will delve into the details of the system's architecture, user roles and functionalities, security measures, user interface design, database management, testing procedures, and plans for future enhancements. We will also highlight the rationale behind the database structure adaptations and the resulting improvements in system performance and data integrity.

2. Project Features

The Primary School Management System encompasses a diverse array of features tailored to the distinct roles within the school ecosystem. These features are designed to enhance operational efficiency, streamline administrative processes, and empower users with tools for effective management. The system offers specialized functionalities for administrators, students, teachers, and staff members.

2.1 Administrator Features

User Management:

Administrators possess the authority to create, modify, and delete user accounts for students, teachers, and staff members. They can assign user roles and control access privileges.

Student Management:

Administrators can manage student profiles, including basic information, enrollment status, and course allocation. They have the capability to enroll students in specific courses and sections.

Teacher Management:

The system allows administrators to manage teacher profiles, assign courses, and oversee grading responsibilities. Teachers are associated with specific courses, and administrators can make course adjustments as needed.

Staff Management:

Administrators can manage staff profiles and their basic details. Staff members have limited access to the system, primarily to view their payment transactions.

Course Management:

Administrators have the authority to define courses, allocate courses to teachers, and enroll students in appropriate courses and sections. Course details such as course codes, titles, and sections can be managed through this feature.

Result Viewing:

Administrators can view and access the results of students, enabling them to monitor academic performance across various courses.

Payment Management:

Administrators can facilitate payment transactions for students, teachers, and staff members. They have an overview of financial transactions and can generate payment reports.

System Overview:

The system provides administrators with an overview of the school's current status, including the number of students, teachers, and staff members enrolled.

2.2 Student Features

Profile Viewing and Update:

Students have access to their profile information and can update personal details as needed.

Result Viewing:

Students can view their academic results and track their performance in various courses.

Course Enrollment:

Students can enroll in courses offered by the school, subject to administrative approval.

Payment History:

Students can view their payment history and transactions, providing transparency into their financial obligations.

2.3 Teacher Features

Profile Viewing:

Teachers can view their profile information and update relevant details.

Grading and Result Entry:

Teachers have the responsibility to grade students' assignments and exams, entering the results into the system.

Result Viewing:

Teachers can access and review the results of students enrolled in the courses they teach.

Transaction History:

Teachers can view their payment transactions, maintaining transparency in financial matters.

2.4 Staff Features

Payment Transaction History:

Staff members can view their payment transactions, ensuring clarity in financial matters.

This section outlines the core features of the Primary School Management System for each user role. It demonstrates the system's capability to cater to the unique needs of administrators, students, teachers, and staff members, ultimately contributing to a well-rounded and efficient management platform.

3. Database Diagram

3.1. Proposed Diagram

Show your initial proposed database diagram.

Table 01: AdminTable

Field	Type	Constraint
adminUsername	Varchar(100)	Default Admin
adminPassword	Varchar(20)	Default Admin
adminPrimaryKey	Composite key	(adminUsername, adminPassword)

Table 02: StudentTable

Field	Type	Constraint
studentId	Int	Primary key

studentFirstName	Varchar (100)	Not null
studentLastName	Varchar (100)	null
studentAge	Int	Not Null
studentFatherName	Varchar (100)	Not null
studentMotherName	Varchar (100)	Not null
studentParentPhone	Varchar (15)	Not null
studentDOB	Date	Not null
studentBG	Varchar (10)	Null
studentAddress	Varchar (200)	Not null
studentCurentClass	Int	Not null
studentSection	Vatrchar(20)	Not null
studentAdmissionDate	Date	Not null
studentGraduationDate	Date	null
studentUsername	Varchar (100)	null
studentPassword	Varchar (20)	null
isCurrentStudent	BIT	Not null

Table 03: TeacherTable

Field	Type	Constraint
teacherId	Int	Primary key
teacherFirstName	Varchar (100)	Not null
teacherLastName	Varchar (100)	null
teacherAge	Int	Not Null
teacherNID	Varchar (100)	Not null
teacherPhone	Varchar (15)	Not null
teacherAddress	Varchar (200)	Not null

teacherBG	Varchar (10)	null
teacherMainCourse	Varchar (100)	Not null
teacherSalary	Money	Not null
teacherJoinDate	Date	Not null
teacerResignDate	Date	Null
teacherUsername	Varchar (100)	Null
teacherPassword	Varchar (20)	Null
isCurrentTeacher	BIT	Not null

Table 04: StaffTable

Field	Type	Constraint
staffId	Int	Primary key
staffFirstName	Varchar (100)	Not null
staffLastName	Varchar (100)	Null
staffAge	Int	Not null
staffNID	Varchar (100)	Not null
staffPhone	Varchar (15)	Not null
staffAddress	Varchar (200)	Not null
staffBG	Varchar (10)	Null
staffJobDesignation	Varchar (100)	Not null
staffSalary	Money	Not null
staffJoinDate	Date	Not null
staffResignDate	Date	Null
staffUsername	Varchar (100)	Null
staffPassword	Varchar (20)	Null

isCurrentStaff	BIT	Not null

Table 05: GradingSyatem

Field	Type	Constraint
studentId	Int	Foreign key
teacherId	Int	Foreign key
courseName	Varchar(20)	Not null
examType	Varchar(20)	Not null
obtainedMark	Decimal(5,2)	Not null
currentClass	Int	Not null
isPassedCourse	BIT	Not null

Table 06: FinalGrade

Field	Type	Constraint
studentId	Int	Foreign key
class1	Varchar(3)	Default null
class2	Varchar(3)	Default null
class3	Varchar(3)	Default null
class4	Varchar(3)	Default null
class5	Varchar(3)	Default null

Table 07: PaymentSystem

Field	Type	Constraint
accountNo	Varchar(20)	Primary key
studentId	Int	Null foreign key
teacherId	Int	Null foreign key
staffId	Int	Null foreign key
paidAmount	Money	Null
restMoney	Money	Null
penaltyAmount	Money	Null
penaltyType	Varchar(100)	Null

Database Relations:

Relation 01: ControlStudent [AdminTable and StudentTable]

Field	Type	Constraint
adminPrimaryKey	Composite key	Foreign key
studentId	int	Foreign key

Relation 02: ControlTeacher [AdminTable and TeacherTable]

Field	Type	Constraint
adminPrimaryKey	Composite key	Foreign key
teacherId	int	Foreign key

Relation 03: ControlStaff [AdminTable and StaffTable]

Field	Type	Constraint

adminPrimaryKey	Composite key	Foreign key
staffId	int	Foreign key

Relation 04: AdminSeeGrade [AdminTable and GradingSystem]

Field	Type	Constraint
adminPrimaryKey	Composite key	Foreign key
studentId	int	Foreign key

Relation 05: AdminSeeFinalGrade [AdminTable and FinalGrade]

Field	Type	Constraint
adminPrimaryKey	Composite key	Foreign key
studentId	int	Foreign key

Relation 06: ControlPayment [AdminTable and PaymentSystem]

Field	Type	Constraint
adminPrimaryKey	Composite key	Foreign key
accountNo	Varchar (50)	Foreign key

Relation 07: Teach [SrudentTable and TeacherTable]

Field	Type	Constraint
studentid	Int	Foreign key
teacherId	Int	Foreign key
courseNameGrading	Varchar(50)	Not Null

Relation 07: StudentSeeFinalGrade [StudentTable and FinalGrade]

Field	Type	Constraint
studentId	Int	Foreign key

Relation 08: StudentPayment [SrudentTable and PaymentSystem]

Field	Type	Constraint
studentid	Int	Foreign key
accountNo	Varchar(50)	Foreign key
studentPaymentDate	Date	Not Null
studentTransactionId	Varchar(50)	Not null
studentTransactionAmmout	Money	Not null
studentTtransactionType	Varchar(100)	Not null
studentPaymentBy	Varchar(20)	Not null

Relation 09: Evaluate [TeacherTable and GradingSystem]

Field	Type	Constraint
teacherId	Int	Foreign key
studentId	Int	Foreign key

Relation 10: TeacherSeeFinalGrade [TeacherTable and FinalGrade]

Field	Type	Constraint
teacherId	Int	Foreign key
studentId	Int	Foreign key

Relation 11: PayTeacher [TeacherTable and PaymentSystem]

Field	Type	Constraint
teacherId	Int	Foreign key
accountNo	Varchar(50)	Foreign key
teacherPaymentDate	Date	Not Null
teacherTransactionId	Varchar(50)	Not null
teacherTransactionAmout	Money	Not null
teacherTransactionType	Varchar(100)	Not null
teacherPaymentBy	Varchar(20)	Not null

Relation 12: PayStaff [TeacherTable and PaymentSystem]

Field	Type	Constraint
staffId	Int	Foreign key
accountNo	Varchar(50)	Foreign key
staffPaymentDate	Date	Not Null
staffTransactionId	Varchar(50)	Not null
staffTransactionAmmout	Money	Not null
staffTransactionType	Varchar(100)	Not null
staffPaymentBy	Varchar(20)	Not null

Limitations

3.1.1. During the initial stages of the project, the proposed database structure included tables such as AdminTable, StudentTable, TeacherTable, StaffTable, GradingSystem, FinalGrade, and PaymentSystem. However, as the project progressed and a deeper analysis was conducted, several limitations were identified in this original design. These limitations primarily revolved around the relationships between tables and the inclusion of the dminTable. To address these limitations, the database structure was revised to achieve greater efficiency, data integrity, and maintainability.

3.1.2. Limitations of Original Table Relationships:

Unnecessary Admin Table:

The original design featured an adminTable with relationships to all other tables, implying that an administrator would have a direct link to every aspect of the system. However, upon further evaluation, it was realized that such an approach would not be efficient or scalable. Granting direct access to an administrator at the database level could compromise data integrity and lead to potential misuse.

Complexity and Overhead:

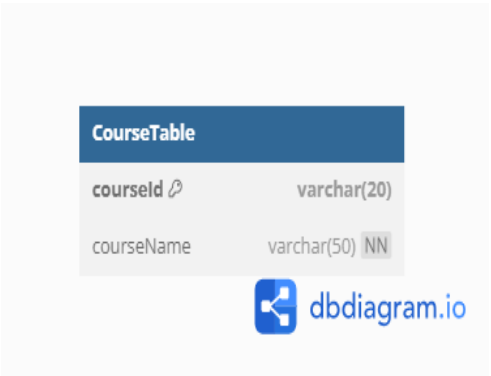
The extensive relationships between the adminTable and other tables added unnecessary complexity and overhead to the system. This would have increased the effort required to manage and maintain the system, potentially leading to performance issues.

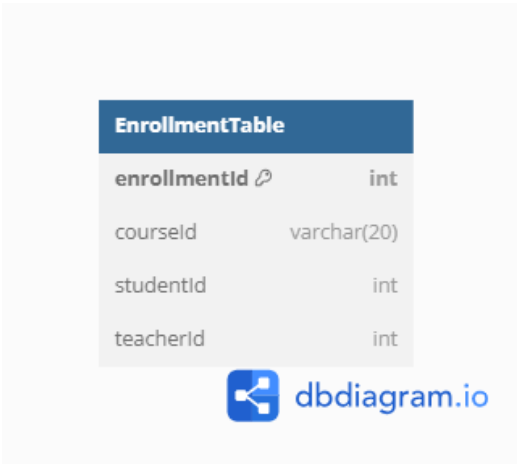
Lack of Data Separation:

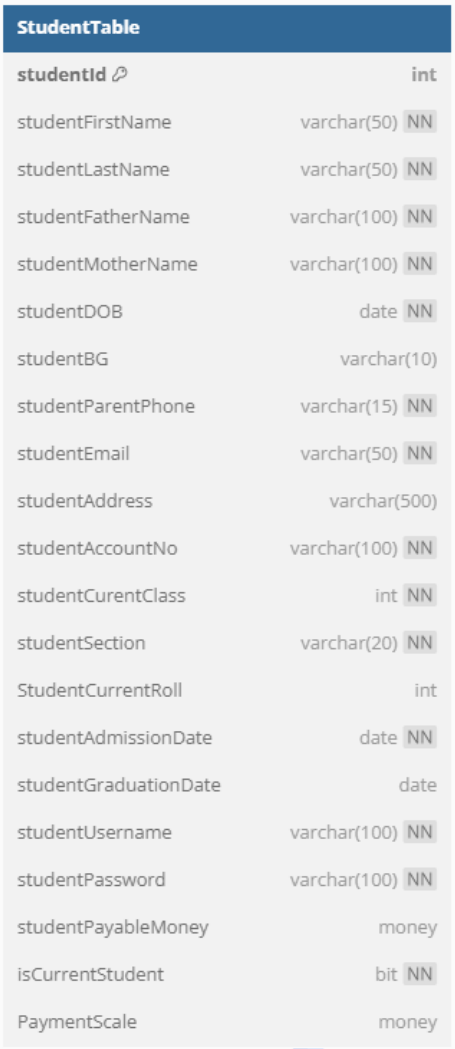
The original design did not provide a clear separation of duties and responsibilities. Having an adminTable with broad access to all other tables could have made it challenging to enforce role-based access control effectively.

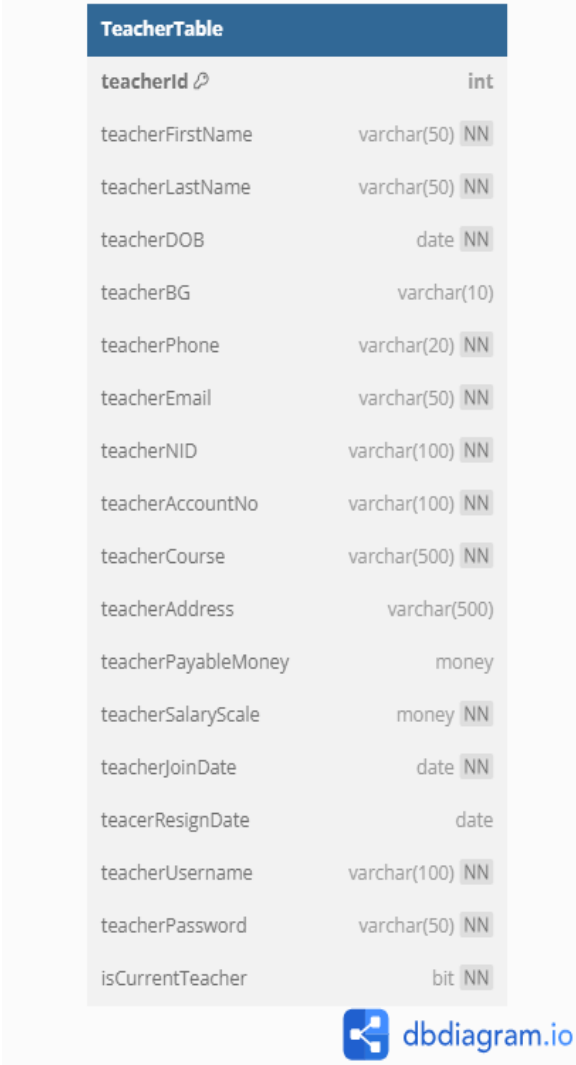
3.2. Final Diagram

Final database diagram[All Tables and Relations]

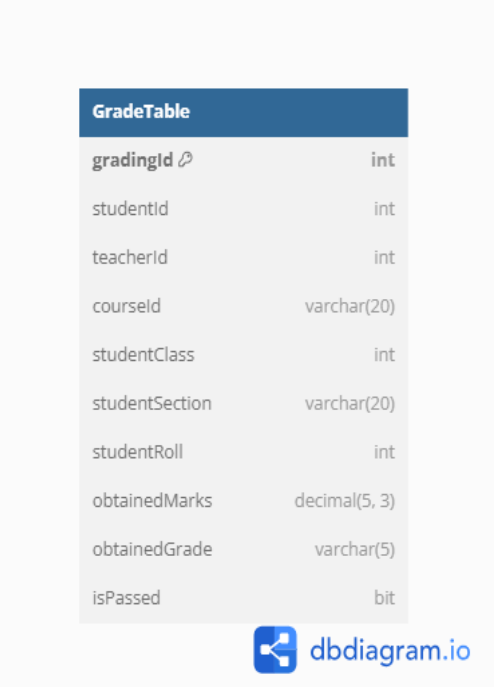


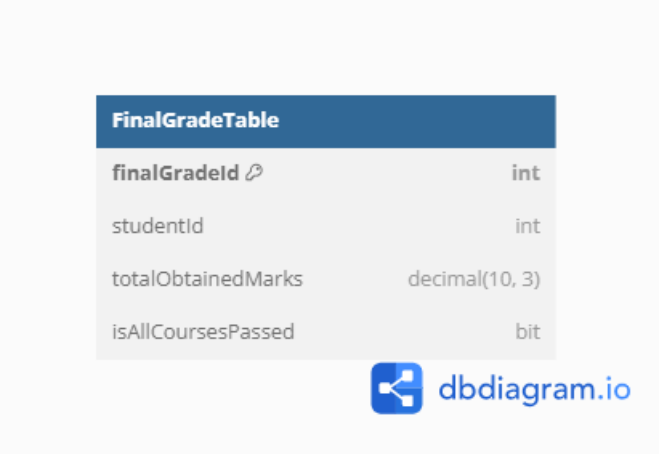


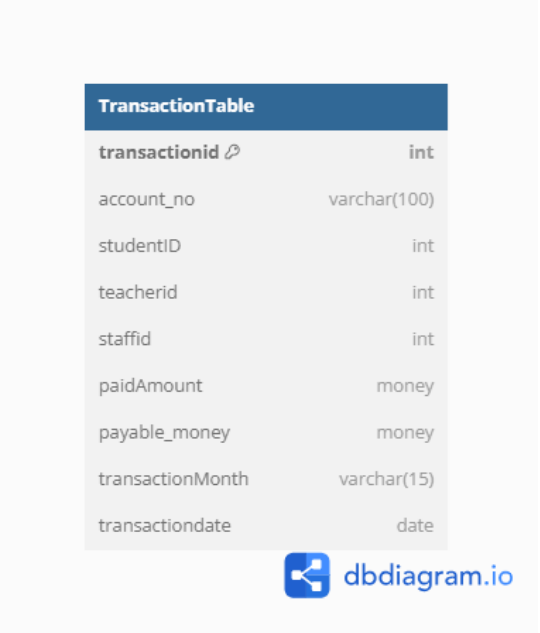

StudentTable
studentId int
studentFirstName varchar(50) NN
studentLastName varchar(50) NN
studentFatherName varchar(100) NN
studentMotherName varchar(100) NN
studentDOB date NN
studentBG varchar(10)
studentParentPhone varchar(15) NN
studentEmail varchar(50) NN
studentAddress varchar(500)
studentAccountNo varchar(100) NN
studentCurentClass int NN
studentSection varchar(20) NN
StudentCurrentRoll int
studentAdmissionDate date NN
studentGraduationDate date
studentUsername varchar(100) NN
studentPassword varchar(100) NN
studentPayableMoney money
isCurrentStudent bit NN
PaymentScale money
dbdiagram.io


TeacherTable
teacherId int
teacherFirstName varchar(50) NN
teacherLastName varchar(50) NN
teacherDOB date NN
teacherBG varchar(10)
teacherPhone varchar(20) NN
teacherEmail varchar(50) NN
teacherNID varchar(100) NN
teacherAccountNo varchar(100) NN
teacherCourse varchar(500) NN
teacherAddress varchar(500)
teacherPayableMoney money
teacherSalaryScale money NN
teacherJoinDate date NN
teacerResignDate date
teacherUsername varchar(100) NN
teacherPassword varchar(50) NN
isCurrentTeacher bit NN
dbdiagram.io


StaffTable
staffId int
staffFirstName varchar(50) NN
staffLastName varchar(50) NN
staffDOB date NN
staffBG varchar(10)
staffPhone varchar(20) NN
staffEmail varchar(50)
staffNID varchar(100) NN
staffAccountNo varchar(50) NN
staffJobDesignation varchar(100) NN
staffAddress varchar(500)
staffPayableMoney money
staffSalaryScale money NN
staffJoinDate date NN
staffResignDate date
staffUsername varchar(100) NN
staffPassword varchar(50) NN
isCurrentStaff bit NN
dbdiagram.io


GradeTable
gradingId int
studentId int
teacherId int
courseId varchar(20)
studentClass int
studentSection varchar(20)
studentRoll int
obtainedMarks decimal(5, 3)
obtainedGrade varchar(5)
isPassed bit
dbdiagram.io


FinalGradeTable
finalGradeId int
studentId int
totalObtainedMarks decimal(10, 3)
isAllCoursesPassed bit
dbdiagram.io


TransactionTable
transactionid int
account_no varchar(100)
studentID int
teacherid int
staffid int
paidAmount money
payable_money money
transactionMonth varchar(15)
transactiondate date
dbdiagram.io

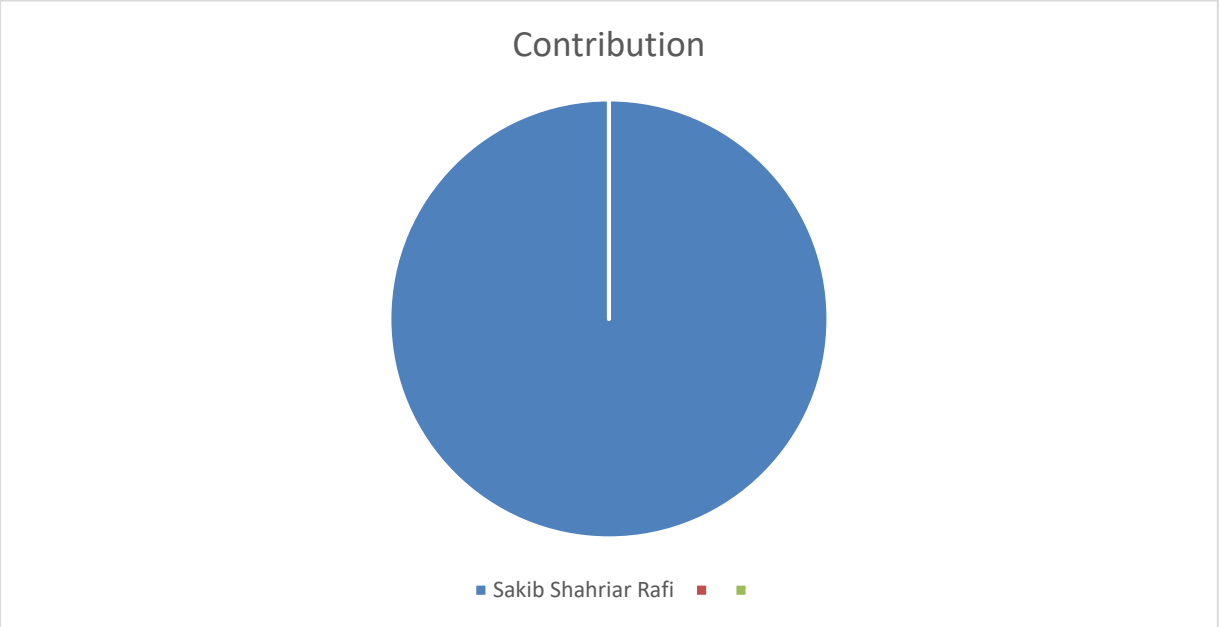
Limitations

3.2.1. One of the pivotal changes in the database design was the deliberate removal of the adminTable. The original concept of a centralized administrative entity with direct database access was reevaluated to address potential complexities and security concerns. Instead, the project team recognized the benefits of managing administrative control at the view level, providing a more streamlined, secure, and intuitive approach.

3.2.2. It can be modified more efficiently in future.

4. Contribution

Sakib Shahriar Rafi's Contribution (just me)



5. Conclusion

In conclusion, the Primary School Management System project stands as a testament to our dedication and technical prowess. Through collaborative efforts, we have successfully developed a comprehensive solution that addresses the needs of various stakeholders in a primary school environment.

The project's robust functionalities, including student and teacher management, result processing, and financial transactions, demonstrate our commitment to creating an efficient and user-friendly system. The meticulous database design and intuitive user interface enhance the system's reliability and accessibility.

As we move forward, we recognize the potential for our system to streamline school operations and contribute to educational efficiency. This project has not only enriched our technical skills but has also deepened our understanding of the role of technology in education.

5.1. Limitations

While the Primary School Management System represents a significant achievement, it is essential to acknowledge its limitations. These limitations stem from various technical and practical challenges that we encountered during the project's development. Recognizing these constraints paves the way for future enhancements and refinements.

Input Validation Challenges

A notable limitation lies in the input validation during the sign-up process. The system currently lacks the ability to detect incorrect phone numbers, email addresses, and National ID (NID) numbers provided during registration. This omission results in erroneous or incomplete data being stored in the database, compromising data integrity.

User Experience with JCalendar
The integration of the JCalendar library presented challenges in terms of user experience. While intended to facilitate date selections, the library's implementation occasionally led to user confusion. A more intuitive approach to date selection could enhance usability and prevent potential input errors.

Data Display and Visualization
The current system's presentation of data could be improved by displaying more comprehensive information from the database. Enhanced data visualization, such as graphs and charts, could provide administrators, teachers, and staff members with clearer insights into various aspects of school management, aiding decision-making processes.

User Interface Design
Although the graphical user interface (GUI) was developed using Java's JFrame framework, further refinements could be made to enhance the overall design aesthetic. Improvements in color schemes, typography, and layout could elevate the user experience and contribute to a more polished appearance.

Limited Testing Due to Small Team
Due to the constraints of a small team and project timeline, thorough testing of the system's functionalities was challenging. Comprehensive testing is crucial to uncovering potential bugs, inconsistencies, or performance issues. A larger testing effort could lead to a more refined and reliable system.

5.2. Future Work
While the Primary School Management System represents a significant achievement, it is essential to acknowledge its limitations. These limitations stem from various technical and practical challenges that we encountered during the project's development. Recognizing these constraints paves the way for future enhancements and refinements.

Advanced Security Measures: Implementing additional security measures, such as two-factor authentication, could further safeguard sensitive data and user accounts.

Automated Notifications: Introducing automated email or SMS notifications for various events, such as exam results or payment reminders, could improve communication within the system.

Intuitive User Guidance: Incorporating tooltips, help sections, or on-screen guidance could assist users in navigating the system effectively, reducing the learning curve.

Extensive Testing: Expanding the testing phase to include a diverse range of scenarios and user interactions would ensure the system's reliability and robustness.

5.3. Project Link
GitHub link: https://github.com/SakibShahriarRafi/Primary-School-Management-System-

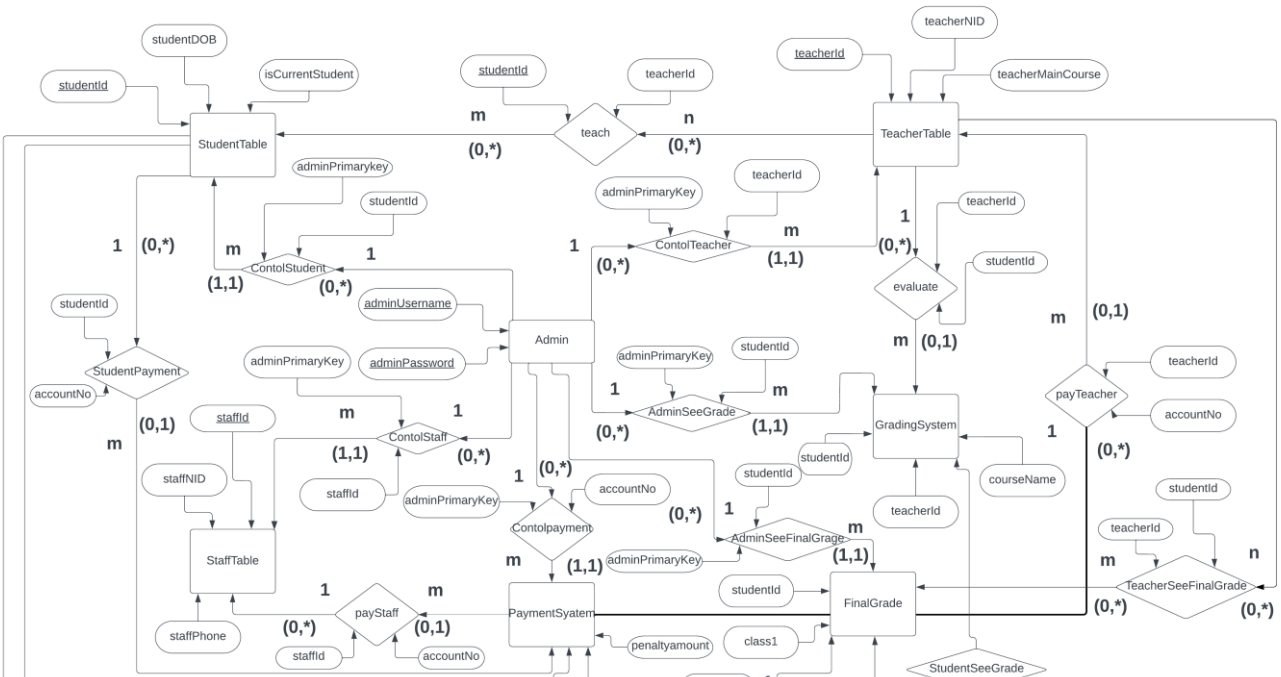


Figure 1: Proposed Diagram

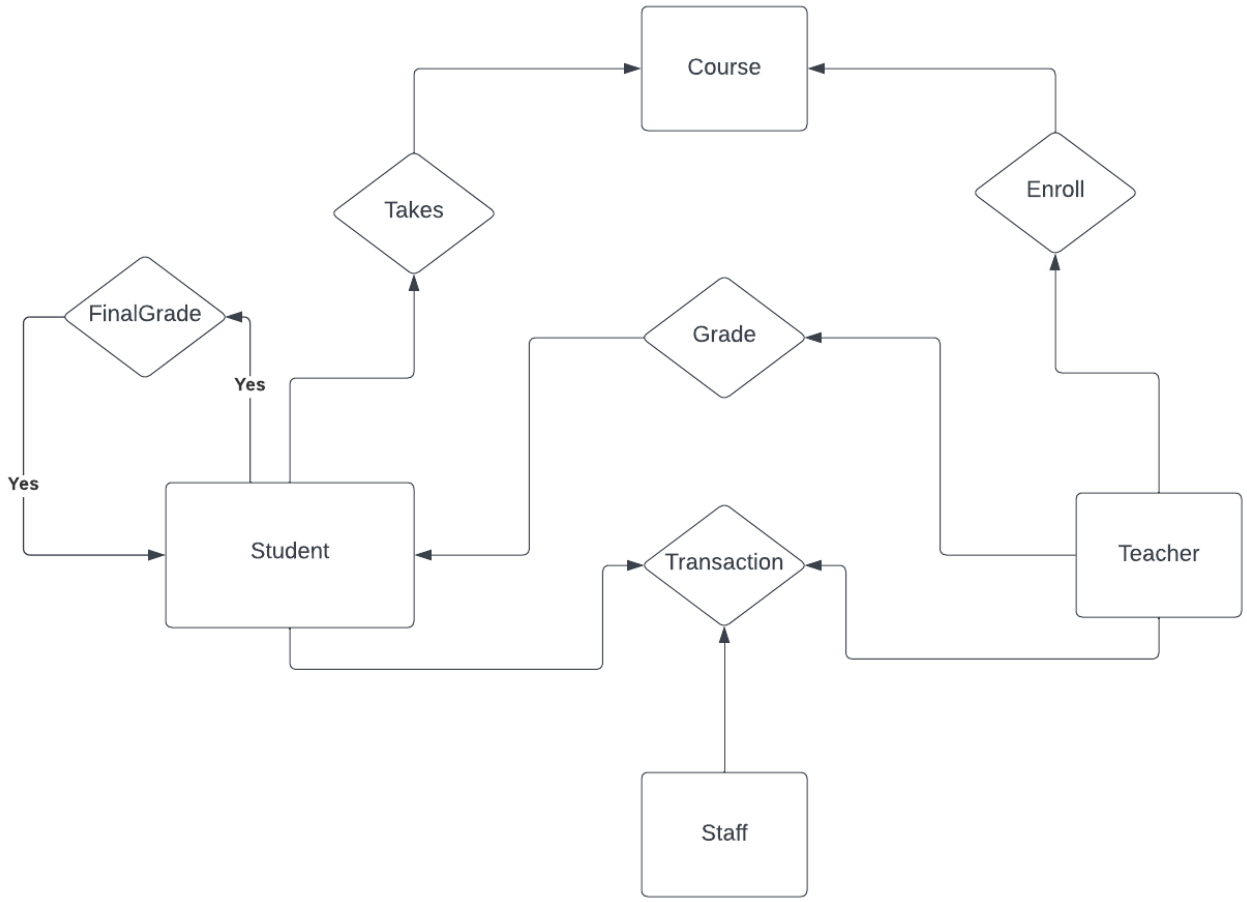


Figure 2: final diagram