School Project

Database class project, academic year 2023/2024

Student Name in English	Student Name in Arabic	Student ID	Section time as shown in zajel	Work percentage
Ala’a Abdelrahim	علاء عبد الرحيم	12112161	11-12	50%
Abdullah Shabib	عبدالله شبيب	12113584	11-12	50%

Date/time

---This section is intended for the Instructor---------------------------------------

Topic	**Mark**
Project Requirements and Modeling	
Correctness of Database mapping	
Functional Dependency and Normalization	
Project Tools	
Project Discussion	
Project Completeness	
Project Output Results or reporting (JasperReport, charts, graphs, etc.)	
Project Administration and Management	
Project Report	
Project Idea	
Project Complexity	
Team work	

Abstract

This report presents the development and implementation of a Database Management System (DBMS) tailored for efficient school administration. The database encompasses key entities such as employees, administrators, classes, subjects, teachers, and students, creating a relational structure for seamless data management.

The Employee table acts as the core, distinguishing between administrators and teachers through the UserType attribute. Admin and Teachers tables store comprehensive information about school staff, including personal details, contact information, and subject assignments. Classes and Subjects tables manage essential class details, such as grade levels, room assignments, and subject information.

The Students table captures vital student information, including personal details, attendance, and transportation preferences. The Teaches_At table establishes the connection between teachers and the classes they instruct, while the Studies table records students' academic performance, linking student IDs with subject names and exam scores.

To maintain data integrity, two triggers have been implemented. The 'trigger_update_studies' ensures that when a teacher is assigned to a class, corresponding study records are created for each student in that class. The 'trigger_create_studies' achieves a similar effect when new students are added to the system.

The database is populated with sample data, demonstrating the system's functionality. Queries and triggers are designed to streamline data management, ensuring accuracy and efficiency in school administration. This DBMS provides a robust foundation for handling various aspects of school management, offering scalability and adaptability to meet evolving educational needs.

Contents:

Introduction

In the ever-evolving landscape of educational institutions, effective management and organization are paramount to ensuring a seamless and efficient learning environment. This report introduces a comprehensive Database Management System (DBMS) designed specifically for the intricate needs of school administration. The system encapsulates the complexities of managing personnel, classes, subjects, and student records, utilizing a relational database model to establish interconnectedness and optimize data retrieval.

In response to the diverse and dynamic demands of modern education, this DBMS is crafted to facilitate the seamless coordination of various facets within a school. From the differentiation of administrators and teachers to the intricate details of class assignments, academic subjects, and student performance, the database acts as a centralized hub for the storage, retrieval, and management of critical information.

This report delves into the intricacies of the database structure, detailing the design choices made to enhance functionality and data integrity. Key entities, including Employees, Admins, Teachers, Classes, Subjects, and Students, are meticulously organized, each playing a crucial role in the overall ecosystem. Furthermore, the report discusses the implementation of triggers to ensure consistent and accurate data representation.

Through the presentation of sample data and a demonstration of query and trigger functionality, this report aims to showcase the versatility and effectiveness of the DBMS in addressing the specific needs of school administration. The system provides not only a comprehensive solution for current management requirements but also a scalable and adaptable framework to accommodate future developments in the educational landscape. As schools continue to embrace technological advancements, this DBMS serves as a foundational tool for fostering streamlined and efficient administrative processes in educational institutions.

Project Requirements

Project Requirements for the School Administration Database Management System:

1. **User Authentication and Authorization:**

- Implement a secure user authentication system to ensure that only authorized personnel can access and modify the database.
- Distinguish between administrators and teachers, each with specific access rights and privileges.

2. **Employee Management:**

- Create an Employee table to store user information, including UserNum, Password, and UserType (Admin or Teacher).
- Include validation checks to ensure data integrity, such as constraints on UserType.

3. **Administrator Management:**

- Develop an Admin table with fields for AdminID, FirstName, LastName, DateOfBirth, Gender, Email, PhoneNumber, and Address.
- Establish a foreign key relationship with the Employee table to maintain data consistency.

4. Classes Management:

- A Classes table must be established to manage class details.
- Fields should include ClassID (primary key), GradeLevel, RoomAlphabet, and DesksNumber.
- GradeLevel should be restricted to values between 1 and 12.

5. Subjects Management:

- A Subjects table is required for storing information about academic subjects.
- Fields should include SubjectName (primary key), Duration, and Description.
- Duration should be restricted to values between 20 and 80.

6. **Teacher Management:**

- Create a Teachers table with fields for TeacherID, FirstName, LastName, DateOfBirth, Gender, Email, PhoneNumber, Address, and SubjectName.

- Establish foreign key relationships with the Employee and Subjects tables for data consistency.

7. **Student Management:**

- Develop a Students table containing information about students, including StudentID, FirstName, MidName, LastName, DateOfBirth, Gender, Address, Parents_Phone_num, Absences, ClassNum, and Transportation.

- Include constraints for gender and transportation choices.

8. **Teaching Assignments:**

- Create the Teaches_At table to establish relationships between teachers and classes they instruct (TeacherID, ClassID).

- Implement foreign key relationships with the Teachers and Classes tables to ensure referential integrity.

9. **Academic Performance Tracking:**

- Design the Studies table to record student academic performance, including StudentID, SubjectName, First_Exam, Second_Exam, and Final_Exam.

- Establish foreign key relationships with the Students and Subjects tables for data consistency.

10. **Triggers for Data Consistency:**

- Implement triggers, such as 'trigger_update_studies' and 'trigger_create_studies,' to automatically update or create records when teachers are assigned to classes or new students are added.

11. **Sample Data Population:**

- Insert sample data into the tables to demonstrate the functionality of the system and assist in testing.

12. **Documentation:**

- Provide comprehensive documentation, including a data dictionary, entity-relationship diagram, and explanations for table structures, triggers, and constraints.

13. **Scalability and Adaptability:**

- Ensure that the database design allows for scalability and adaptability to meet future educational needs and changes in the school's administrative processes.

By meeting these project requirements, the School Administration Database Management System will serve as a robust and efficient tool for managing various aspects of school administration.

Functional Dependency

1. Employee Table:
   UserNum -> password, UserType.

2. Admin Table:
   AdminID -> FirstName, LastName, DateOfBirth, Gender, Email, PhoneNumber, Address
   Email -> FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Address
   PhoneNumber -> FirstName, LastName, DateOfBirth, Gender, Email, Address

3. Teachers Table:
   TeacherID -> FirstName, LastName, DateOfBirth, Gender, Email, PhoneNumber, Address, SubjectName
   Email -> FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Address, SubjectName
   PhoneNumber -> FirstName, LastName, DateOfBirth, Gender, Email, Address, SubjectName

4. Students Table:
   StudentID -> FirstName, MidName, LastName, DateOfBirth, Gender, Address, Parents_Phone_num, Absences, ClassNum, Transportation

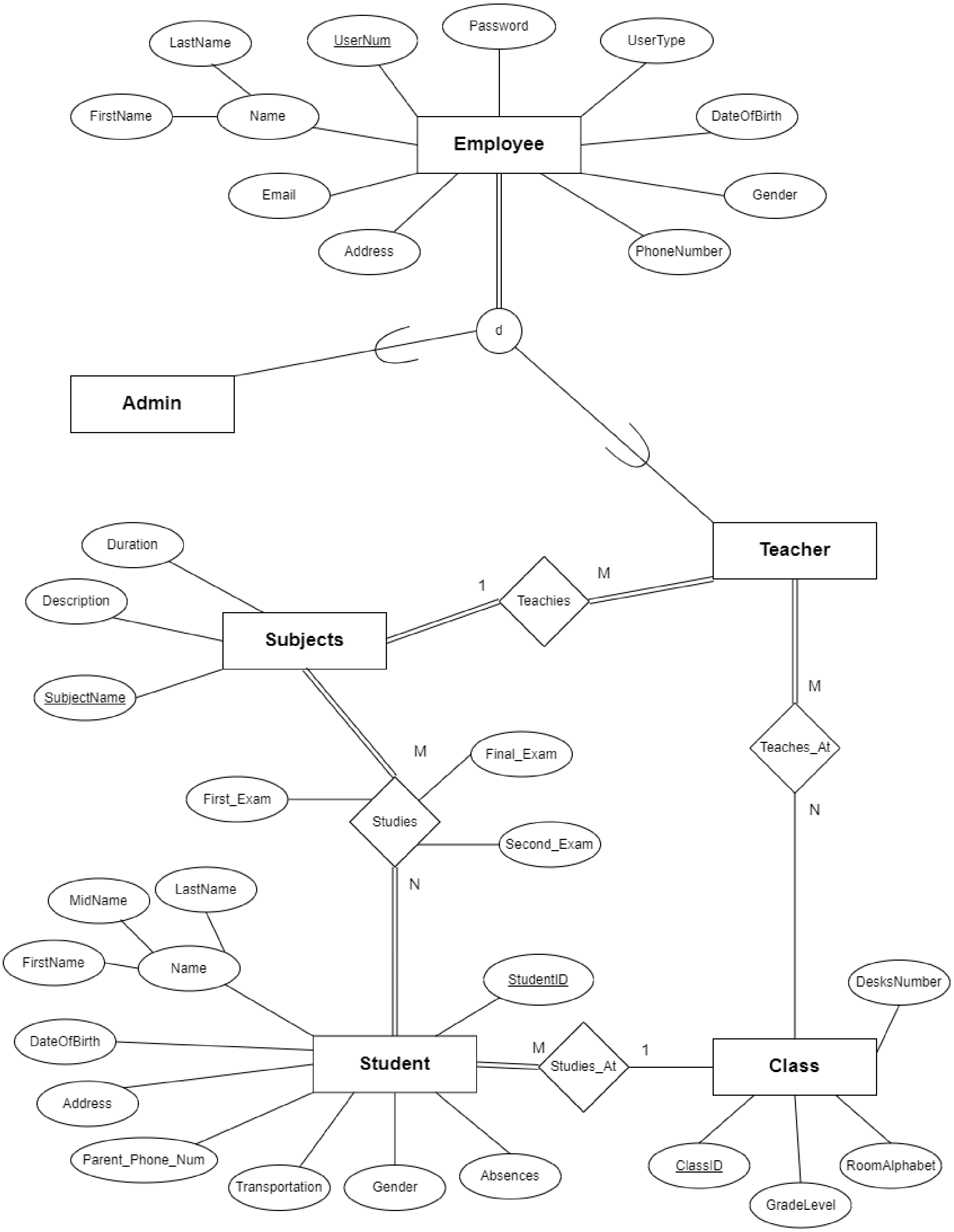
5. Classes Table:
   ClassID -> GradeLevel, RoomAlphabet, DesksNumber

6. Subjects Table:
   SubjectName -> Duration, Description

7. Teaches_At Table:
   (TeacherID, ClassID) -> (TeacherID, ClassID)

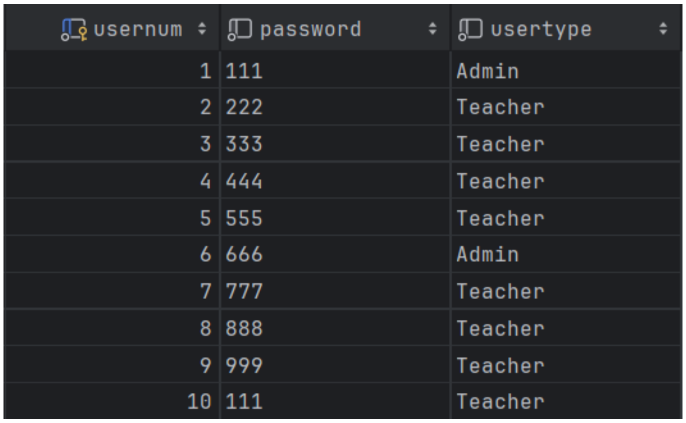
8. Studies Table:
   (StudentID, SubjectName) -> First_Exam, Second_Exam, Final_Exam

Project EER Model



Normalization Process

1. Employee Table:



usernum	password	usertype
1	111	Admin
2	222	Teacher
3	333	Teacher
4	444	Teacher
5	555	Teacher
6	666	Admin
7	777	Teacher
8	888	Teacher
9	999	Teacher
10	111	Teacher

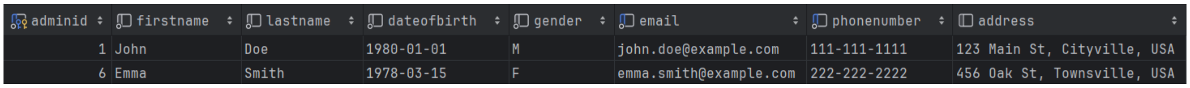
The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and there is only one candidate key (UserNum).

The table is in 3NF as it is already in 2NF, and there are no transitive dependencies.

The table is in BCNF since there is only one candidate key (UserNum) and no non-trivial dependencies on superkeys.

2. Admin Table:



adminid	firstname	lastname	dateofbirth	gender	email	phonenumber	address
1	John	Doe	1980-01-01	M	john.doe@example.com	111-111-1111	123 Main St, Cityville, USA
6	Emma	Smith	1978-03-15	F	emma.smith@example.com	222-222-2222	456 Oak St, Townsville, USA

The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and there is only one candidate key (AdminID).

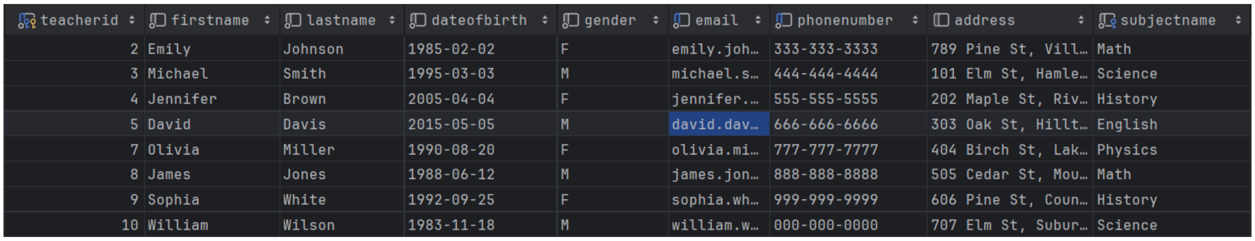
The table fails in 3NF

We observe that there is a transitive dependency between Email and PhoneNumber on one side and FirstName, LastName, DateOfBirth, Gender on the other side. To eliminate this transitive dependency, we decompose the table:

AdminID | FirstName | LastName | DateOfBirth | Gender | Address

Email | PhoneNumber | AdminID

3. Teachers Table:



teacherid	firstname	lastname	dateofbirth	gender	email	phonenumber	address	subjectname
2	Emily	Johnson	1985-02-02	F	emily.joh…	333-333-3333	789 Pine St, Vill…	Math
3	Michael	Smith	1995-03-03	M	michael.s…	444-444-4444	101 Elm St, Hamle…	Science
4	Jennifer	Brown	2005-04-04	F	jennifer.…	555-555-5555	202 Maple St, Riv…	History
5	David	Davis	2015-05-05	M	david.dav…	666-666-6666	303 Oak St, Hillt…	English
7	Olivia	Miller	1990-08-20	F	olivia.mi…	777-777-7777	404 Birch St, Lak…	Physics
8	James	Jones	1988-06-12	M	james.jon…	888-888-8888	505 Cedar St, Mou…	Math
9	Sophia	White	1992-09-25	F	sophia.wh…	999-999-9999	606 Pine St, Coun…	History
10	William	Wilson	1983-11-18	M	william.w…	000-000-0000	707 Elm St, Subur…	Science

The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and there is only one candidate key (TeacherID).

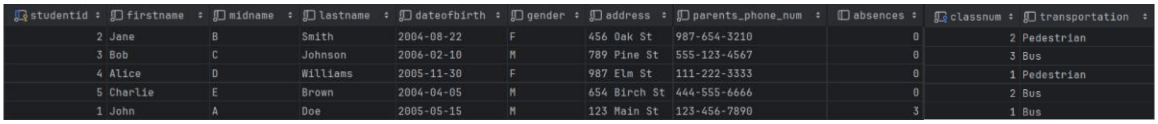
The table fails in 3NF

We observe that there is a transitive dependency between Email and PhoneNumber on one side and FirstName, LastName, DateOfBirth, Gender on the other side. To eliminate this transitive dependency, we decompose the table:

TeacherID | FirstName | LastName | DateOfBirth | Gender | Address | SubjectName

Email | PhoneNumber | TeacherID

4. Students Table:



studentid	firstname	midname	lastname	dateofbirth	gender	address	parents_phone_num	absences	classnum	transportation
2	Jane	B	Smith	2004-08-22	F	456 Oak St	987-654-3210	0	2	Pedestrian
3	Bob	C	Johnson	2006-02-10	M	789 Pine St	555-123-4567	0	3	Bus
4	Alice	D	Williams	2005-11-30	F	987 Elm St	111-222-3333	0	1	Pedestrian
5	Charlie	E	Brown	2004-04-05	M	654 Birch St	444-555-6666	0	2	Bus
1	John	A	Doe	2005-05-15	M	123 Main St	123-456-7890	3	1	Bus

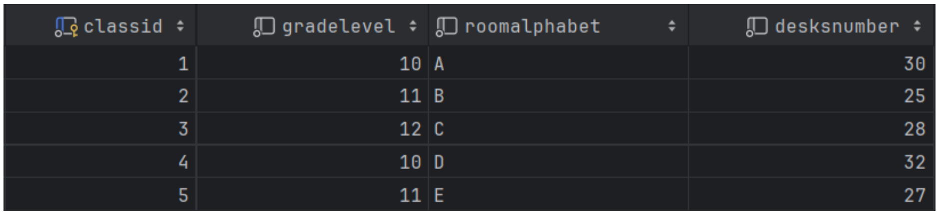
The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and there is only one candidate key (StudentID).

The table is in 3NF as it is already in 2NF, and there are no transitive dependencies.

The table is in BCNF since there is only one candidate key (StudentID) and no non-trivial dependencies on superkeys.

5. Classes Table:



classid	gradelevel	roomalphabet	desksnumber
1	10	A	30
2	11	B	25
3	12	C	28
4	10	D	32
5	11	E	27

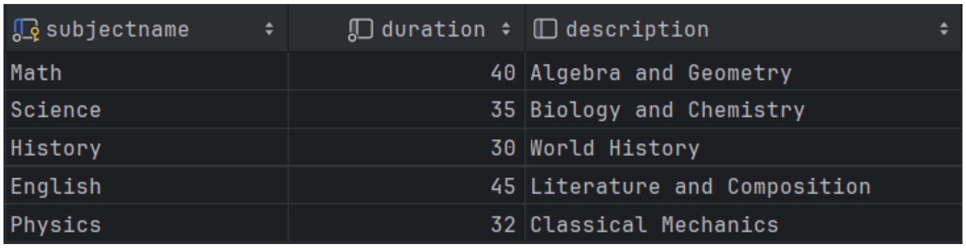
The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and there is only one candidate key (ClassID).

The table is in 3NF as it is already in 2NF, and there are no transitive dependencies.

The table is in BCNF since there is only one candidate key (ClassID) and no non-trivial dependencies on superkeys.

6. Subjects Table:



subjectname	duration	description
Math	40	Algebra and Geometry
Science	35	Biology and Chemistry
History	30	World History
English	45	Literature and Composition
Physics	32	Classical Mechanics

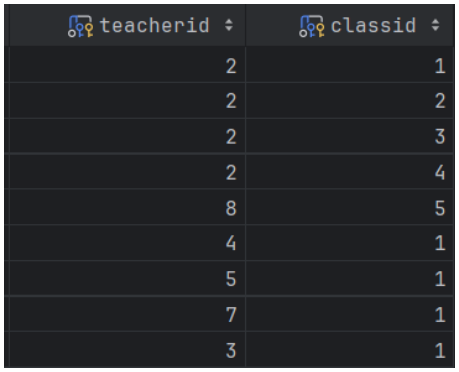
The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and there is only one candidate key (SubjectName).

The table is in 3NF as it is already in 2NF, and there are no transitive dependencies.

The table is in BCNF since there is only one candidate key (SubjectName) and no non-trivial dependencies on superkeys.

7. Teaches_At Table:



teacherid	classid
2	1
2	2
2	3
2	4
8	5
4	1
5	1
7	1
3	1

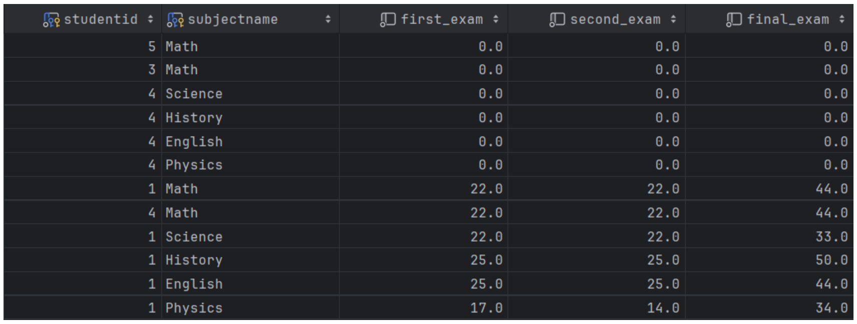
The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and the primary key is composite.

The table is in 3NF as it is already in 2NF, and there are no transitive dependencies.

The table is in BCNF since there is only one candidate key ((TeacherID, ClassID)) and no non-trivial dependencies on superkeys.

8. Studies Table:



studentid	subjectname	first_exam	second_exam	final_exam
5	Math	0.0	0.0	0.0
3	Math	0.0	0.0	0.0
4	Science	0.0	0.0	0.0
4	History	0.0	0.0	0.0
4	English	0.0	0.0	0.0
4	Physics	0.0	0.0	0.0
1	Math	22.0	22.0	44.0
4	Math	22.0	22.0	44.0
1	Science	22.0	22.0	33.0
1	History	25.0	25.0	50.0
1	English	25.0	25.0	44.0
1	Physics	17.0	14.0	34.0

The table is in 1NF as it contains only atomic values, and there are no repeating groups.

The table is in 2NF as it is already in 1NF, and the primary key is composite.

The table is in 3NF as it is already in 2NF, and there are no transitive dependencies.

The table is in BCNF since there is only one candidate key ((StudentID, SubjectName)) and no non-trivial dependencies on superkeys.

Tools

We utilized these tools and websites to aid us in project completion.

- SQL:
  - PostgreSQL
  - DataGrip 2023.2.3
- Diagram:
  - Draw.io
- GUI:
  - Apache NetBeans IDE 19
- Reports:
  - Jaspersoft Studio-6.20.6
- Logo, Icons and Images:
  - www.iconfinder.com (icons)
  - www.flaticon.com (icons)
  - www.logo.com (logo)
  - www.istockphoto.com (images)

Discussion

We will delve into each interface, exploring the functionalities of each:

1. Login_Page:

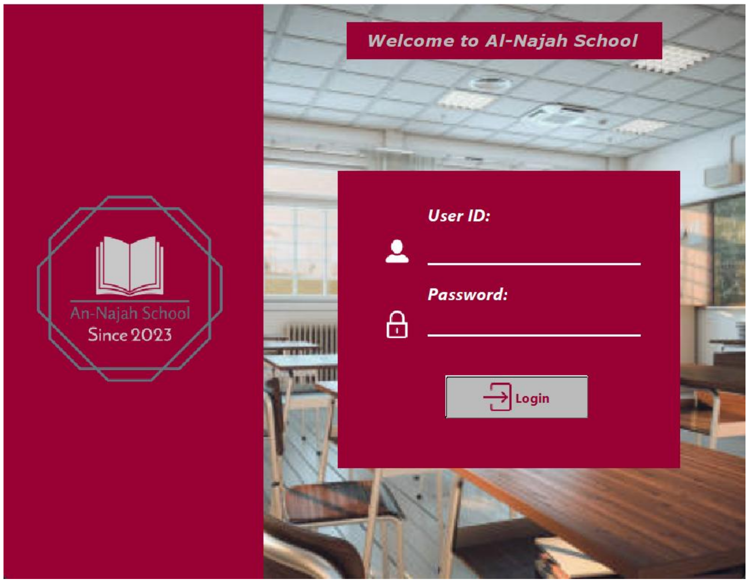


Figure 1: Login_Page interface.

This interface is configured at the program's initiation. Upon entering their ID and password, users are redirected to specific interfaces based on their roles as either administrators or teachers. Prior to this redirection, the system verifies the user's role to ensure they are directed to the appropriate interface.

We have two interfaces: one designed for administrators and another for teachers. We will address each interface separately, examining its unique features and functionalities.

Admin Interfaces

Let's begin by discussing the Admin Interface:

2. Admin_Home_Page:

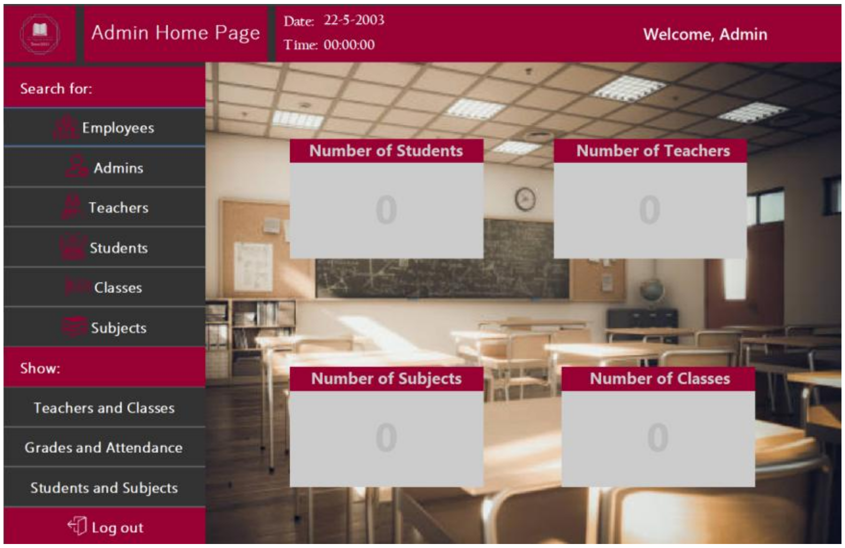


Figure 2: Admin_Home_Page interface.

This Interface is intended for administrators. Upon entering their credentials on the Login_Page, administrators gain access to functionalities that allow them to control aspects of the database system not accessible to other users.

This interface displays various details, including the admin's name and the number of students in the database. Additionally, it features several buttons that empower administrators to make modifications to the database.

3. Admin_Employees:

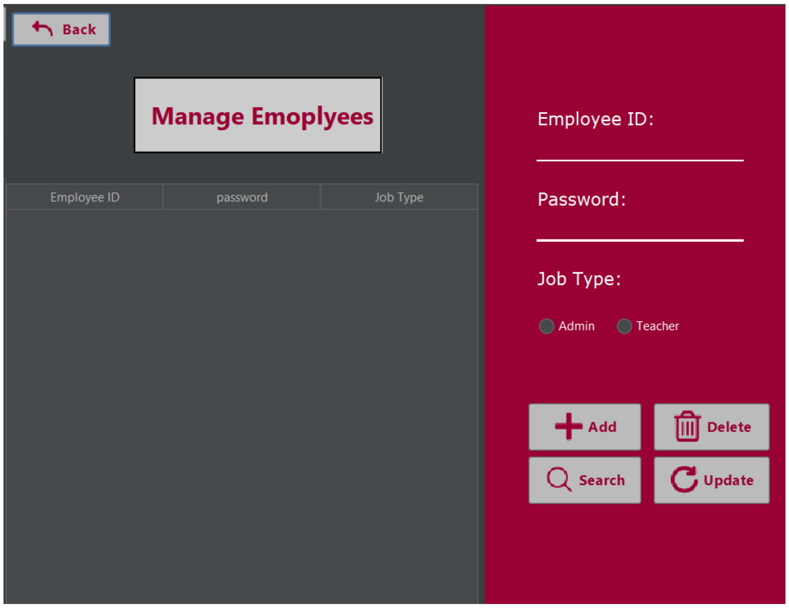


Figure 3: Admin_Employees Interface.

Access to this interface is granted to the admin by selecting the "Employees" button. This action directs the admin to the functionalities associated with employee management within the system.

Within this interface, administrators have the capability to add a new employee. This is achieved by completing the required text fields and selecting appropriate options from the radio buttons, followed by pressing the "Add" button.

To remove an employee, the administrator must input the respective Employee ID in the designated field and then proceed to use the delete button. This action ensures the targeted employee is deleted from the system.

For updating employee information, the admin is required to enter the specific Employee ID to be updated. Subsequently, the admin must populate each field with the necessary information and conclude the process by pressing the "Update" button. This ensures that the employee's details are appropriately modified in the system.

To perform a search, the admin is prompted to input the Employee ID in the designated field and then initiate the search by pressing the corresponding button. The output will be displayed in the text fields and radio buttons, providing the admin with the relevant information.

4. Admin_Admins:

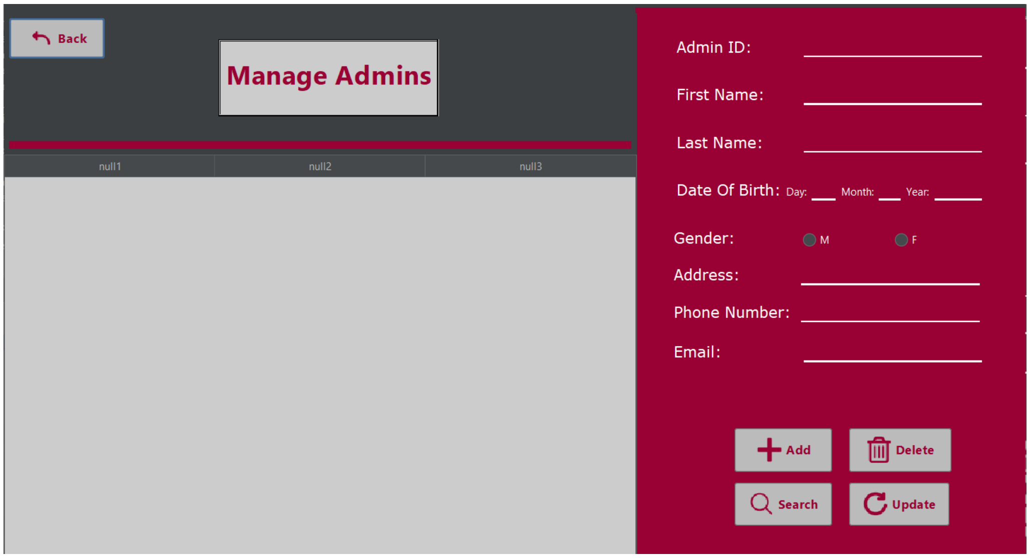


Figure 4: Admin_Admins interface.

Access to this interface is given to the admin upon choosing the "Admins" button. This action directs the admin to functionalities associated with the management of other admins within the system. This interface shares similar functionalities with the Admin_Employees Interface.

5. Admin_Teachers:

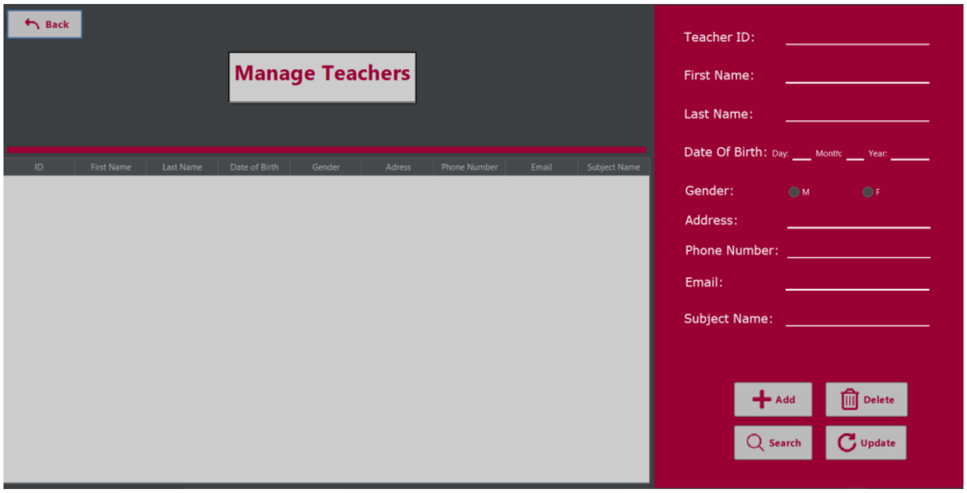


Figure 5: Admin_Teachers interface.

Access to this interface is given to the admin upon choosing the "Teachers" button. This action directs the admin to functionalities associated with the management of Teachers within the system. This interface shares similar functionalities with the Admin_Employees Interface.

6. Admin_Students:

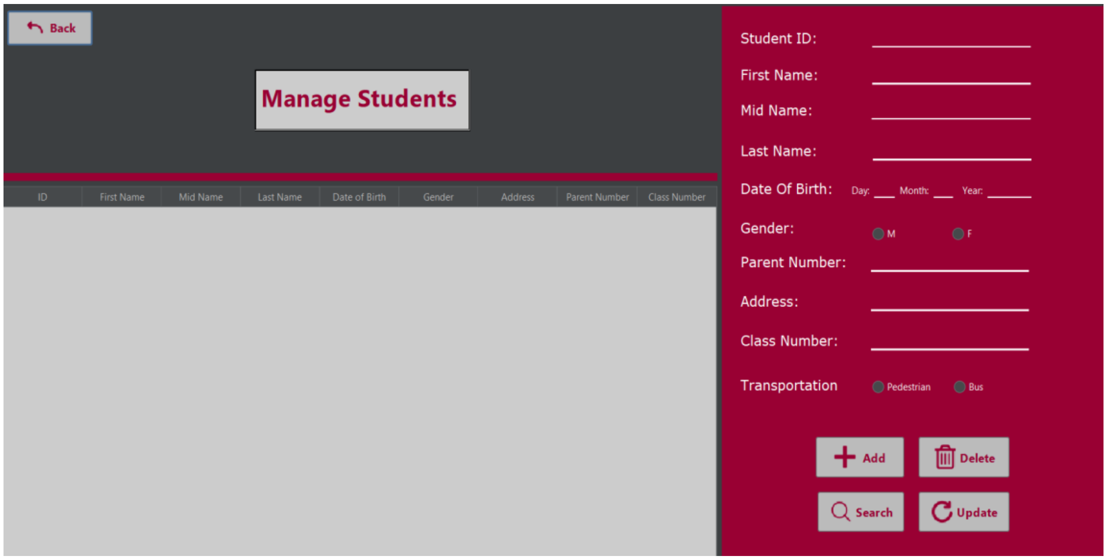


Figure 6: Admin_Students interface.

Access to this interface is given to the admin upon choosing the "Students" button. This action directs the admin to functionalities associated with the management of Students within the system. This interface shares similar functionalities with the Admin_Employees Interface.

7. Admin_Classes:

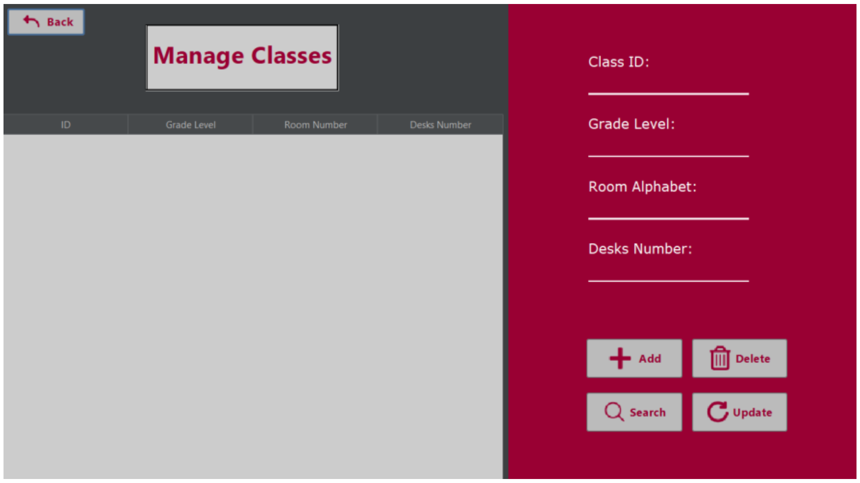


Figure 7: Admin_Classes interface.

Access to this interface is given to the admin upon choosing the "Classes" button. This action directs the admin to functionalities associated with the management of Classes within the system. This interface shares similar functionalities with the Admin_Employees Interface.

8. Admin_Subjects:

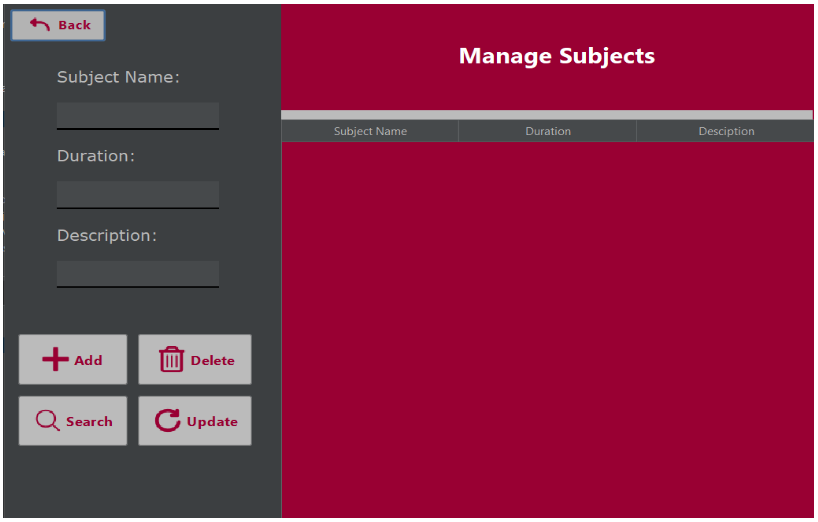


Figure 8: Admin_Subjects interface.

Access to this interface is given to the admin upon choosing the "Subjects" button. This action directs the admin to functionalities associated with the management of Subjects within the system. This interface shares similar functionalities with the Admin_Employees Interface.

9. Classes_and_Teachers:

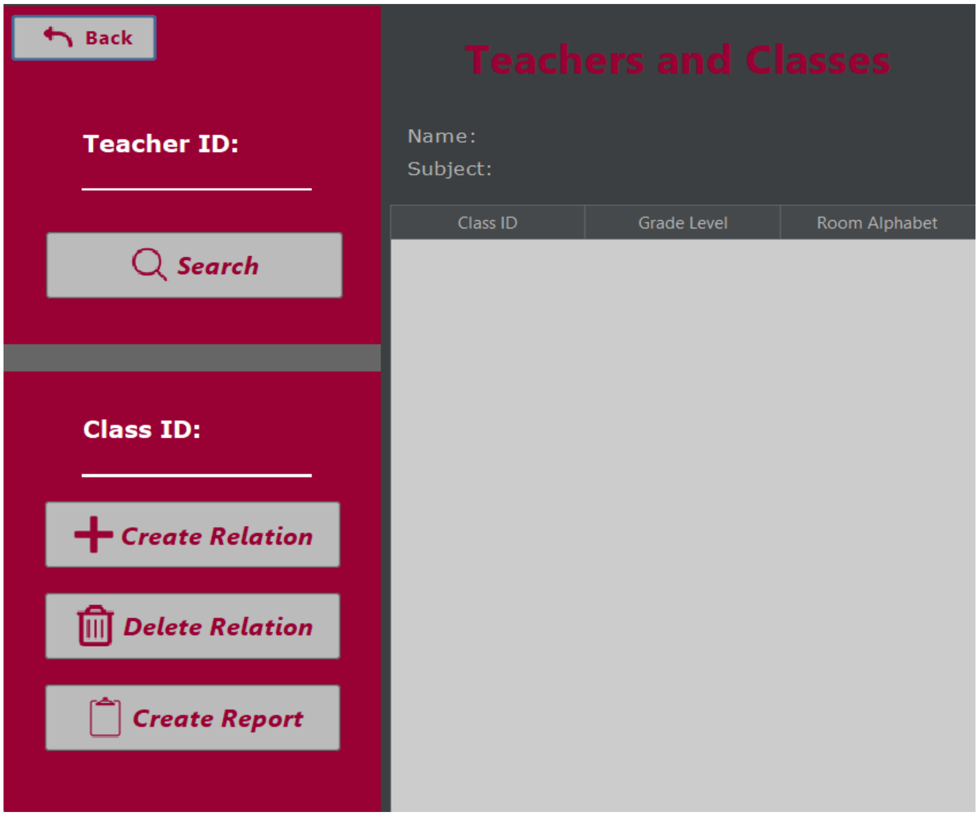


Figure 9: Classes_and_Teachers interface.

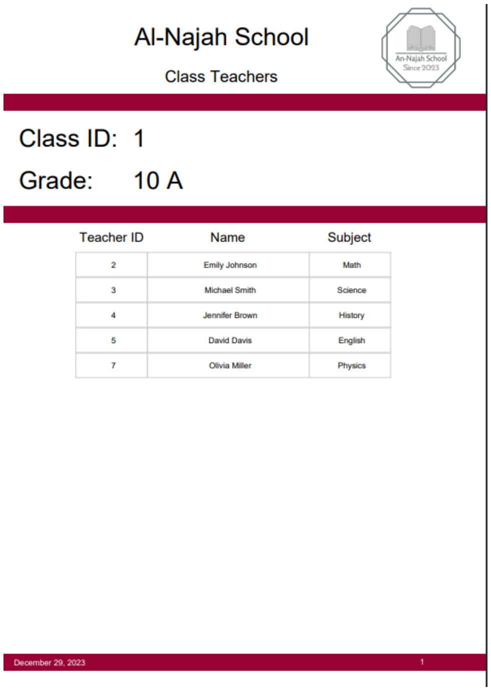
Access to this interface is given to the admin upon choosing the "Teachers and Classes" button.

Within this interface, the admin has the capability to establish relationships between teachers and classes. For each teacher, relevant information such as their name, the subject they teach, and the classes they are assigned to can be displayed. This feature facilitates efficient management of teacher-class associations.

To establish a relation, the admin needs to complete the necessary fields with the required information and then proceed to press the "Create Relation" button. This action will generate a connection between the students and the teacher’s subject.

To Delete a relation, the admin needs to complete the necessary fields with the required information and then proceed to press the "Delete Relation" button.

The "Create Report" button serves the purpose of generating a report for the class identified by the "Class ID" field. This report encompasses all teachers who instruct in that particular class.



Al-Najah School

Class Teachers

Class ID: 1

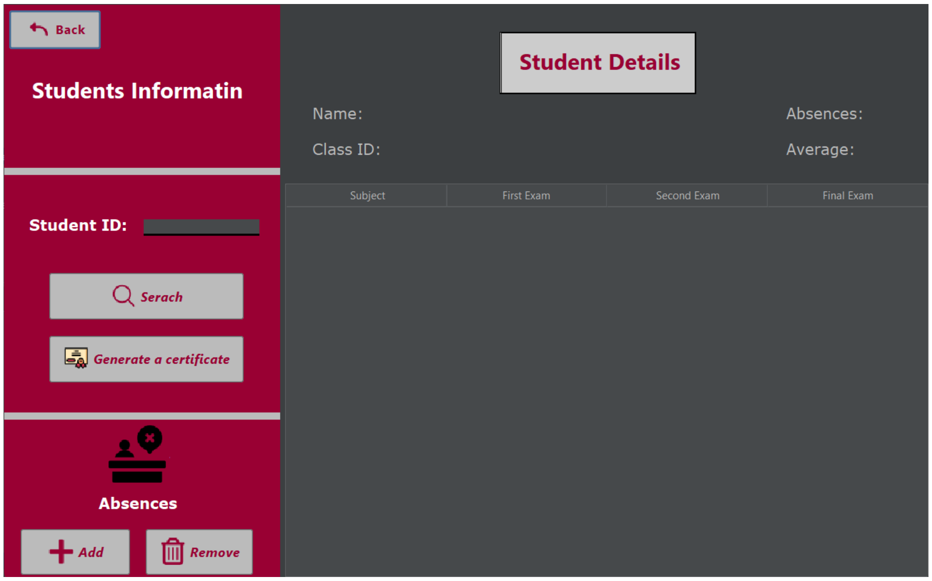
Grade: 10 A

Teacher ID	Name	Subject
2	Emily Johnson	Math
3	Michael Smith	Science
4	Jennifer Brown	History
5	David Davis	English
7	Olivia Miller	Physics

December 29, 2023 1

Figure 10: Jasper Report for Class ID = 1.

10. Search_for_Students_Info:



Student ID:

Name:

Class ID:

Absences:

Average:

Subject	First Exam	Second Exam	Final Exam

Figure 11: Search_for_Students_Info interface.

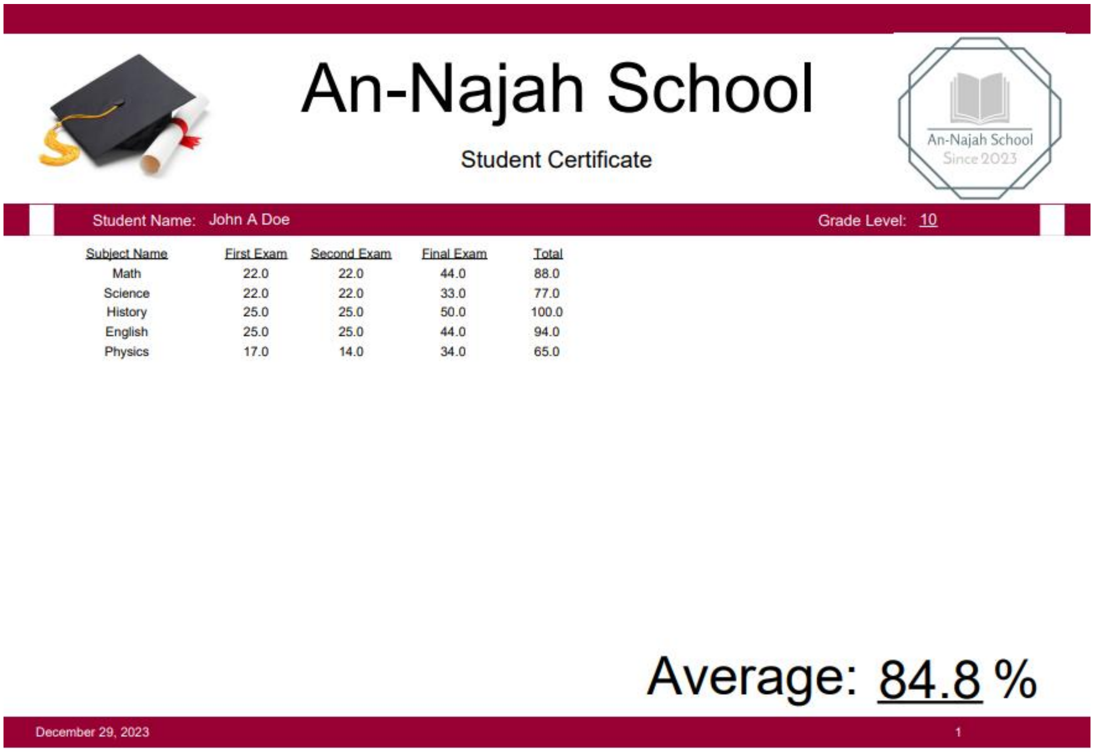
Access to this interface is given to the admin upon choosing the "Grades and Attendance" button.

In this interface, the admin can see the student’s first and last name, his class ID, absences, the average and the subjects with its marks by searching by his ID

Within this interface, the admin is provided with the ability to view specific details about a student by searching using their ID. The displayed information includes the student's first and last name, class ID, absences, average, and subjects along with their respective marks. This search functionality allows for a comprehensive overview of an individual student's academic and attendance details.

The "Generate Certificate" button is specifically designed to create a certificate for the student whose ID is entered into the required field. Upon activation of this button, the system will generate the certificate in the form of a PDF file for the specified student.

The "Add" and "Remove" buttons are configured to respectively increase or decrease the number of absences for a student. These actions provide a convenient means for the administrator to manage and update attendance records.



An-Najah School

Student Certificate

Student Name: John A Doe

Grade Level: 10

Subject Name	First Exam	Second Exam	Final Exam	Total
Math	22.0	22.0	44.0	88.0
Science	22.0	22.0	33.0	77.0
History	25.0	25.0	50.0	100.0
English	25.0	25.0	44.0	94.0
Physics	17.0	14.0	34.0	65.0

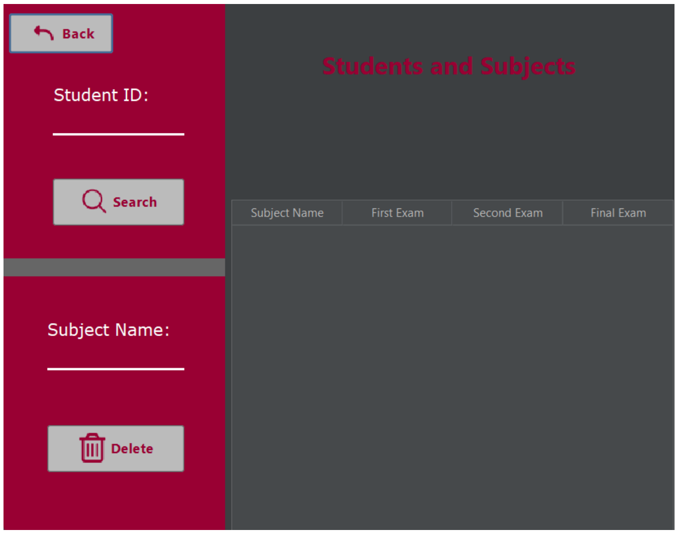
Average: 84.8 %

December 29, 2023

1

Figure 12: Jasper Report For Student ID = 1.

11. Students_and_Subjects:



Subject Name	First Exam	Second Exam	Final Exam

Figure 13: Students_and_Subjects interface.

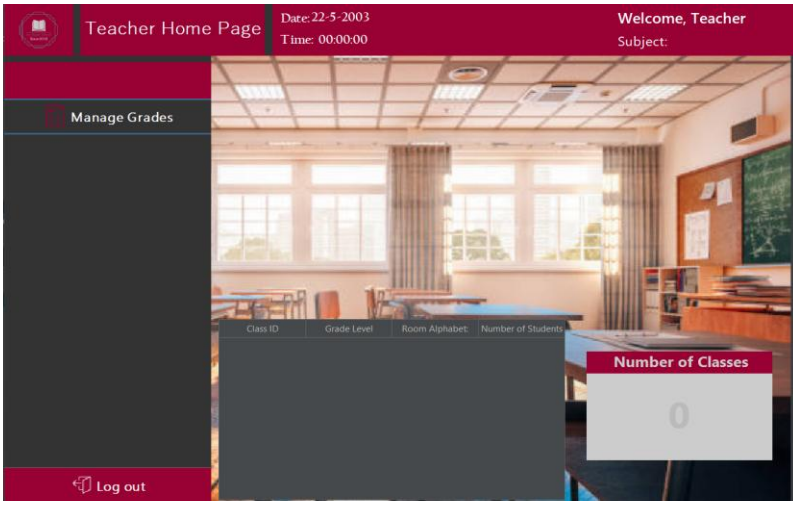
Access to this interface is given to the admin upon choosing the "Students and Subjects" button.

This interface is designed to check the relationships between students and subjects. It involves entering the student's ID into the designated field, allowing for a specific examination of the connections between the student and the subjects they are associated with.

The "Delete" button is utilized to remove or delete the existing relationship between a student and a subject. Pressing this button initiates the process of deleting the specified relation.

Teacher Interfaces

12. Teacher_Home_Page:



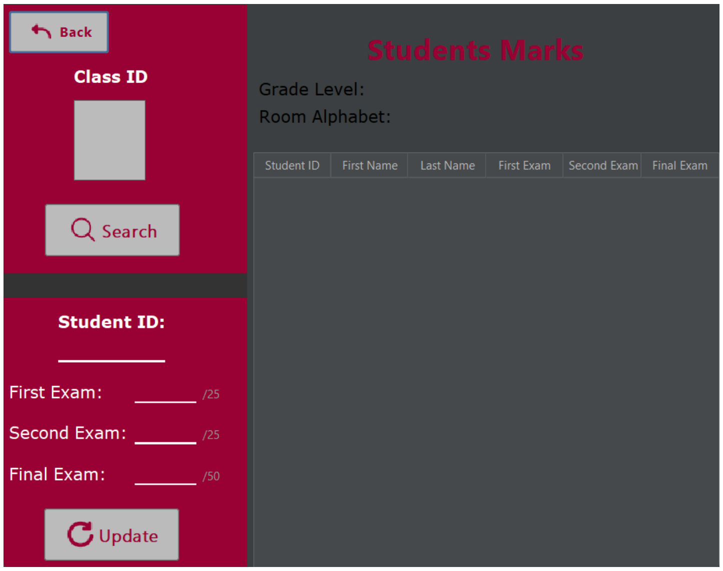
Date: 22-5-2003
Time: 00:00:00
Subject:

Class ID	Grade Level	Room Alphabet	Number of Students

This interface is specifically designed for teachers, accessible after entering the User ID and password on the login page.

Within this interface, teachers can access information such as their name, the subjects they teach, and the classes they are assigned to. This provides teachers with an overview of their teaching responsibilities.

13. Teacher_Students_Marks:



Student ID	First Name	Last Name	First Exam	Second Exam	Final Exam

Access to this interface is given to the teacher upon choosing the "Manage Grades" button.

The teacher has the option to choose the class they teach from the Class ID list, and subsequently search for details such as its level, alphabet, and the list of students enrolled in that specific class. This functionality enables teachers to retrieve relevant information about the class and its students.

Teachers have the ability to update the marks of students by filling in the Student ID field for a particular class. Subsequently, they can input the marks for exams in the designated fields. This allows teachers to efficiently manage and update the academic records of students in a given class.

Conclusion

The establishment and implementation of the school administration Database Management System (DBMS) presented in this database schema signify a comprehensive solution for managing various facets of educational institutions. The design, encompassing tables for employees, administrators, classes, subjects, teachers, and students, aims to provide an efficient and structured platform for handling critical administrative data.

The adherence to normalization principles, including Boyce-Codd Normal Form (BCNF), ensures that the database maintains data integrity and minimizes redundancy. The normalization process, albeit not explicitly illustrated, underscores the importance of a well-thought-out schema, resulting in tables that are inherently in compliance with BCNF conditions.

The inclusion of triggers, such as 'update_studies_on_teacher_assignment' and 'create_studies_on_student_insert,' demonstrates a forward-thinking approach to automation within the system. These triggers enhance the database's functionality by automating the creation of study records, streamlining processes related to teacher assignments and student enrollments.

The provided sample data showcases the practical utility of the database, offering insights into its adaptability and effectiveness in handling diverse scenarios within a school setting. The inclusion of administrators, teachers, and students, each with their respective details and relationships, exemplifies the versatility of the system.

In summary, the school administration DBMS serves as a foundational tool for educational institutions seeking to leverage modern technology for efficient and organized administrative processes. Its well-defined structure, adherence to normalization principles, and automation features through triggers collectively position it as a valuable asset for schools looking to enhance their administrative capabilities in the dynamic landscape of education.

References

The information utilized in the project is derived from the following references:

- Database Slides
- ChatGPT 3.5
- YouTube