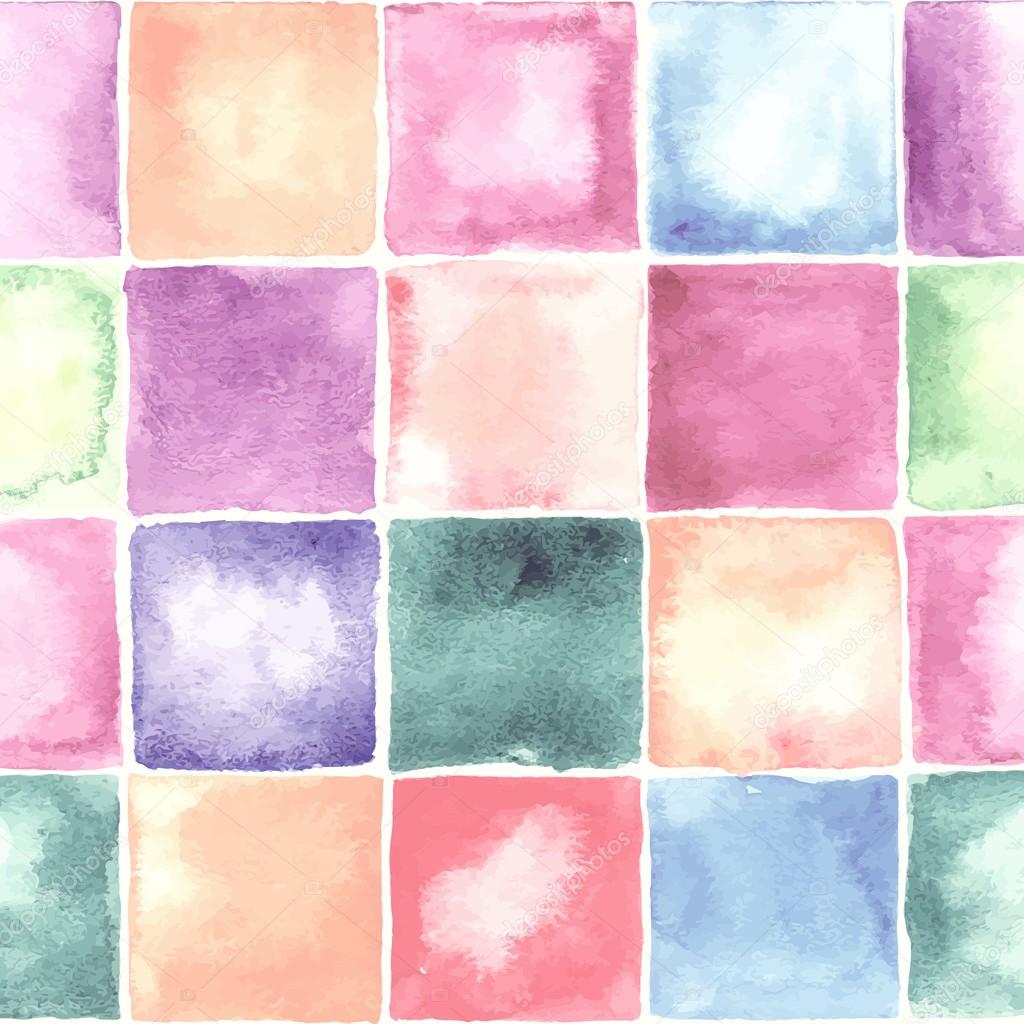
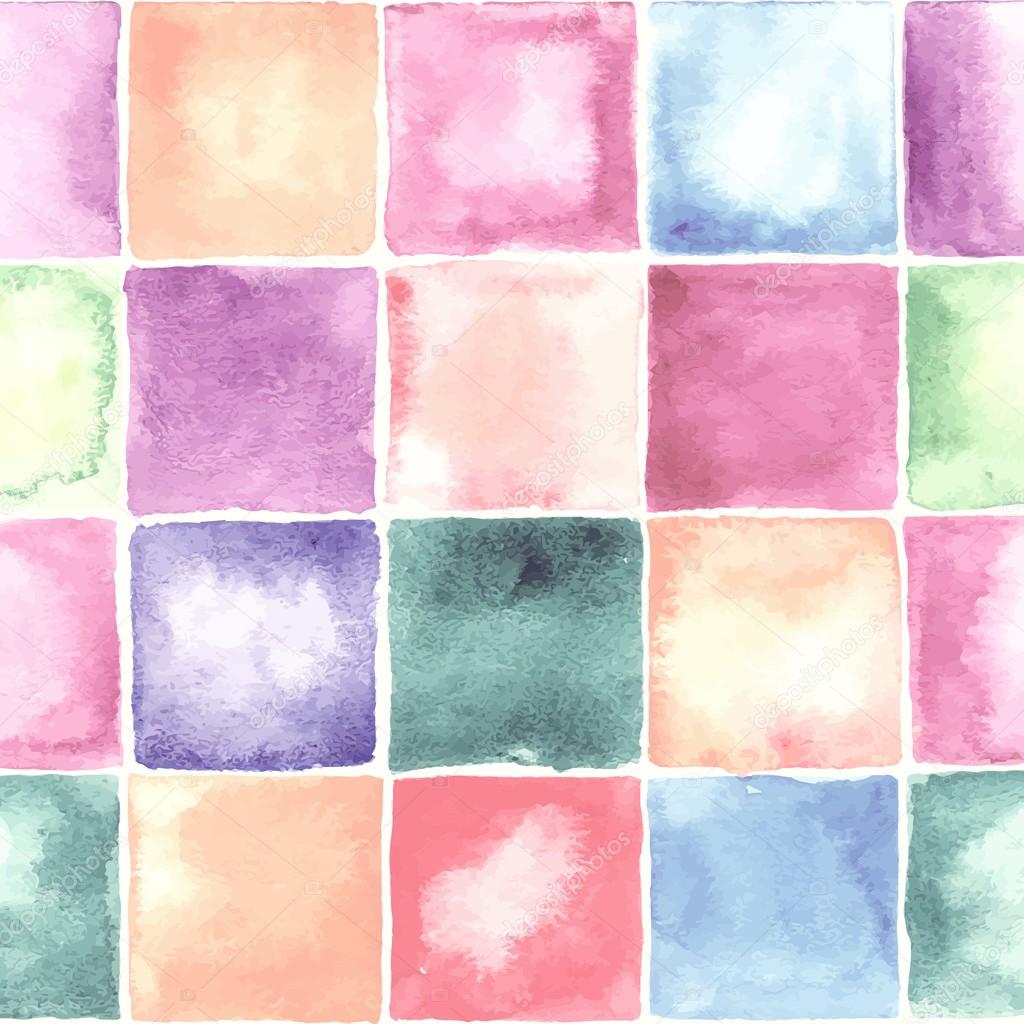


Logo

Description automatically generated

International University - Principles of Database Management

Principles of Database Management

GROUP PROJECT 1

Instructor: Assoc. Prof. Nguyen Thi Thuy Loan

Topic 4: Solution for Saving Student Records

ITDSIU19060

ITITIU19252

ITDSIU19017

ITITIU19059

ITITIU20177

Nguyễn Ái Vương

Nguyễn Quí Vinh Quang

Nguyễn Duy Minh Thông

Trương Công Trung

Lê Đỗ Minh Đăng

**Member contributions:**

|  |  |  |
| --- | --- | --- |
| **Member** | **Contribution Rate** | **Tasks** |
| Nguyen Qui Vinh Quang | 33% | Design ERD, create the schema, build backend, design UI, and write a report |
| Nguyen Ai Vuong | 22% | Collect requirements, draw ERD, design UI, and design Slide |
| Nguyen Duy Minh Thong | 22% | Create the schema, draw ERD, design UI, and design Slide |
| Truong Cong Trung | 22% | Collect requirements, build backend, and write the report |
| Le Do Minh Dang | 1% | Join the group chat |

**CONTENT TABLE**

I – Introduction

II – System and Design Specification

1. Entity Relationship Diagram
2. Schema

III – Implementation

1. Database
2. Application

**Case 1: You are a student**

**Case 2: You are a teacher**

**Case 3: You are an admin**

IV – Conclusion

I – Introduction

The education industry is gradually developing strongly in an era where people must live with the current covid-19 pandemic. Typically, many universities have expanded their enrollment after adopting online learning.

Moreover, each student has many properties in recent years, such as saving digital information in the record. Because big data needs an efficient algorithm, we decided to implement this project.

In this project, our teams aim to build a database that saves a student record at the university, such as a student's name, student id, the score of each subject, and so on. Our team aims for the database to be able to divide the information in the form of tables from which users can query information easily and quickly.

II – System and Design Specification

1. Entity Relationship Diagram

Our team proposes the ER diagram to overview our database in the first step.Diagram, schematic

Description automatically generated

II.A 1

* The table Student store the essential student information. It contains basic information for everyone to know a specific student, such as first name, last name, birthday, and mail. The academic year attribute can have everyone know how long a student was in university. We also let a student has the **StudentId**as a primary key to speed up the querying process in applications that utilize this database.
* The table Semester billing record of the money flow that students need no know. It will store the bill that students have to pay each semester. Every semester, the university will update this table. Semester billing has the following attributes:

**+ Year**: This attribute helps us know the year that the bill was created. The year attribute can have values like 2020 and 2021.

**+ Semester:** This attribute helps us know the semester that the bill was created. The year attribute can have values like 2020 and 2021.

**+ Price per credit**: This attribute stores students' fees for every credit they register. This price is calculated based on the cost of the dollar at a specific time.

**+ Insurance price**: Every student has to pay health insurance and accident insurance. This attribute stores the sum of both these insurances.

* The table Scholarship will store all the scholarships that students have. Scholarship has the following attributes:

+ **Year**: This is a primary key. This attribute gives the information on which years a student received a scholarship.

+ **Semester**: This is a primary key. This attribute gives the information on which semester a student received a scholarship.

**+ Tuition**: This attribute gives the total money that a student can receive from this scholarship.

**+ Scholarship Description:** This attribute describes what the scholarship is.

* User: This table will store the password and the User Role in the system.

**+ Username:** This is the primary key and has a unique value.

**+ Password**: This attribute will store the hashed password of the user account to increase the security robustness of our database.

**+ Role**: This attribute helps us determine the user's role: admin, teacher, or student.

* Administrator: To simplify this database, our team decided that the administrator table will store all the user information who has permission to edit this database. This table has the following attributes:

**+ Admin id**: The unique value is a primary key.

**+ Admin name**: This attribute stores the information about the administrator's name.

* Course: This table stores all the subjects which the university has. This table is the crucial table that every student record database must have. This table has the following attributes:

**+ Course Id**: The unique value of the subject is a primary key.

+ **Course name**: The name of the course.

**+ Credits**: This attribute stores the total credits of the course.

**+ Description**: This attribute indicates what the subject will teach and all the information relative to it.

* Teacher: Because this database is about saving student records, our team stores some fundamental attributes about each lecturer in this table. Attributes of this table:

**+ Teacher Id**: Each teacher's unique value will be a specific id.

**+ Teacher name**: Name of a teacher.

**+ Department**: This attribute describes a particular academic discipline of a teacher.

* Class: At the beginning of each semester, an administrator will create a class. The class is a combination of the teacher and course. In this table, it stores some attributes:

**+ Teacher Id**

**+ Course Id**

**+ Section**: This attribute shows us the day in the week and time that subject has. This attribute is multivalued because we can have some classes on different days and times.

**+ Size**: This attribute gives us information about the maximum student this class can have.

**+ Inclass percentage, midterm percentage, and final percentage**: these three attributes represent the weight of each score column.

* Enrolled Class: This table is a table that stores student records. This table shows the connection between Student and Class. This table has an attribute score that shows the subject's student performance.

1. Schema

We construct the Relational Schema based on the ER diagram; the schema is given below:

Diagram

Description automatically generated

Figure: II.B 1: Relation Model

Diagram

Description automatically generated

Figure: II.B 2: The diagram database in the SQL server

III – Implementation

1. Database

Student saving record was first created following precisely what the ERD and the relational diagram show. Our team used basic SQL to initialize the database. We also put the code to initialize into a folder named ‘createDB.sql.’ In the beginning, our data must insert some records about the admin account and admin information. After that, our team also insert some values about student information, teacher information, and course information. This information can still be added to the application.

1. Application

Graphical user interface, application

Description automatically generated

Figure: II.2.1: Log in UI

When you open our application, you have to log in to your account. Each username will show that your role is a student, a teacher, or an administrator. For each role, the app will open the corresponding window.

**Case 1: You are a student**

Graphical user interface, application

Description automatically generated

Figure: II.2.2: Log in UI

Graphical user interface, application

Description automatically generatedWhen you Log in success, the first display you see looks like figure II.2.2. You can update your personal information like first name, last name, and email. Or you also can update your password account. The table on the right of the display shows your study record history.

Figure: II.2.3: Course Registration

We also have the windows for student registration II.2.3. At this window, the student can choose all the possible courses to open next semester on the left table. The right table lists all the classes for that the student was registered.

Graphical user interface, application

Description automatically generatedStudents can also keep track of their school tuition fees. This table will show all the costs that students had to pay for each semester.

Figure: II.2.4: Student billing

**Case 2: You are a teacher**

When you Log in success, the first display you see looks like figure II.2.5. From this display, a teacher can choose their current teaching class and list all the students in that class. The teacher also can update their score

Graphical user interface

Description automatically generated

Figure: II.2.5: Teacher's main view

**Case 3: You are an admin**

As an administrator, you can do all the things that students and teachers can do. You can also add a new student, a new teacher, and a new course. You can assign a class for a teacher to teach. (figure II.2.6, figure II.2.7, figure II.2.8, and figure II.2.9)

Graphical user interface, application

Description automatically generated

Figure: II.2.6: Assign a class for a teacher.

Table

Description automatically generated

Figure II.2.8: The administrator can edit student information, add a new student, and remove a student from a database.

Graphical user interface, application

Description automatically generated

Figure: II.2.7: The administrator can edit teacher information, add a new teacher, and remove a teacher from a database.

Graphical user interface, application

Description automatically generated

Figure II.2.9: The administrator can edit course information, add a new course, and also remove a course from a database.

Graphical user interface, table

Description automatically generatedAdmin can also edit the tuition for each semester. It means that the administrator can set the price per credit and insurance price. Figure 9.

Figure II.2.10: Tuition information

IV – Conclusion

From the project, we achieved a deeper understanding of building an application and how to connect an application to a database. We also see that the first design is essential.