



华南理工大学  
South China University of Technology

# Database System Concepts

## Experiment 2 Report

(Program Manual & Design Document)

Student Name: Jevin Laudo

Student Number: 202069990040

Class: CST2020

Teacher: 陈俊龙、汪秀敏、余志文

## Contents

Abstract .....	3
Program Design .....	4
SQL Server Database Setup .....	4
Project Solution Setup .....	4
SQL Connection Configuration .....	5
Initial SQL Database Table Setup .....	6
Sample Data Insertion .....	7
UI Design with Windows Forms .....	9
Program Manual .....	14
UI Components and Layout .....	14
Login Page .....	14
Student Portal .....	16
Teacher Portal .....	16
Admin Portal .....	17
Query_PersonalInfo .....	18
Query_CourseInfo .....	20
Query_ScoreLookup .....	20
Query_AverageScore .....	21
Modify_Score .....	23
Insert_Student .....	25
Insert_Teacher .....	26
Insert_Course .....	27
Insert_Choose .....	28
Modify_Student .....	29
Modify_Teacher .....	29
Modify_Course .....	30
Modify_Choose .....	31
Delete_Student .....	32
Delete_Teacher .....	32
Delete_Choose .....	33
Conclusion .....	34

## **SCUT Management Information System**

Management information systems (MIS) are the backbones of most organizational operations today. It is a tool to gather and analyse data, as well as to report information gathered as an aid in decision-making.

Moreover, these systems are also utilized throughout universities across the globe, generally as an essential component in the management of academic activities and internal affairs. This project aims to simulate and develop a functional management information system for the college of Computer Science at the South China University of Technology, where academic-related information is stored and managed through the usage of SQL database, a relational database management system.

This paper covers the program design, program manual, database design, general UI design and components utilized, as well as SQL commands utilized to achieve the desired result.

## Program Design

The program is developed with Microsoft Visual Studio 2022, with the .NET graphical class library Windows Forms, which is responsible for managing the Graphical User Interface (GUI) of the project. Functionalities of the program is written in the C# programming language, whereas data is stored, retrieved, and managed through Microsoft SQL Server 2019 version 15.0.2000.5.

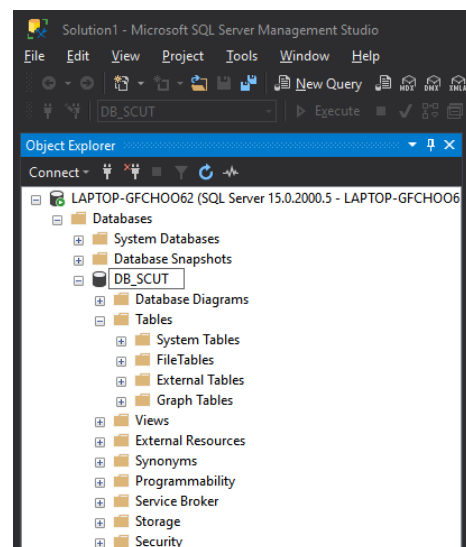
### SQL Server Database Setup

The initial pre-development stage of the program design is to construct the SQL Server database that the program will interact with. This stage is done with the aid of SQL Server Management Studio (SSMS), where connections to SQL Server database through localhost will be configured.

As observed in figure 1, a new database has been created named “DB\_SCUT”, where information for the MIS will be stored within.

**Figure 1**

*Newly Generated SQL Server Database*

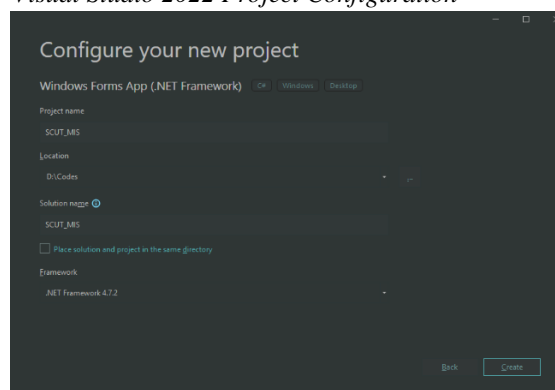


### Project Solution Setup

Within Microsoft Visual Studio IDE, the project is configured as a Windows Forms app, running on the .NET framework with C# as its programming language.

**Figure 2**

*Visual Studio 2022 Project Configuration*



Within the project solution, the file `app.config` contains the configuration setup made for the project, including the .NET version support, as well as the connection strings for the program to connect to the SQL Server database.

**Figure 3**

*Visual Studio 2022 app.config Project Configuration*

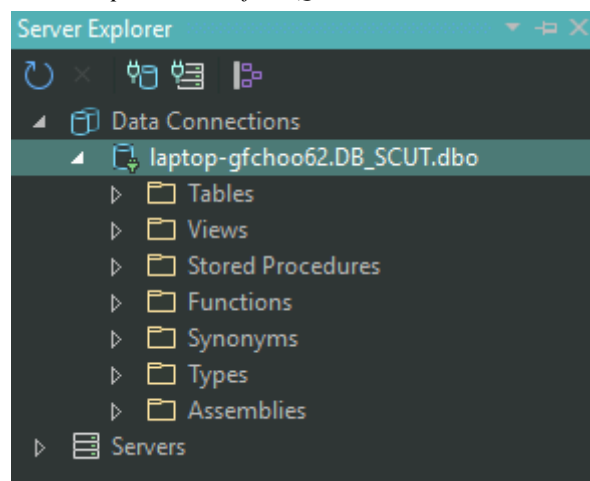
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="database"
          connectionString="Data Source=LAPTOP-GFCH0062;
                           Initial Catalog=DB_SCUT;
                           Integrated Security=True"
          providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
</configuration>
```

### **SQL Connection Configuration**

As presented in figure 3, the SQL database connection string for the project is preconfigured and is contained within the `<connectionStrings>` tags. The string will be utilized within the project whenever the SQL database is accessed. This approach is chosen for the project implantation to promote better maintainability, as well as ease of usage.

**Figure 4**

*Server Explorer View for SQL Database Connection*



## Initial SQL Database Table Setup

**Core system information table construction.** Following the requirements, four tables are used within the system. Among which are the students' information table, teachers' information table, courses information table, as well as the course choosing table. `students`, `teachers`, `courses`, and `choose` are the names of the tables respectively.

**Figure 5**

*Core System Information Table Creation Commands*

```
CREATE TABLE students(
    sid VARCHAR(10) NOT NULL PRIMARY KEY,      -- Student ID
    sname NVARCHAR(20) NOT NULL,                -- Student Name
    sex VARCHAR(6) CHECK(sex = 'male' or sex = 'female') NOT NULL, -- Student Sex
    entrance_age INT CHECK(entrance_age >= 10 and entrance_age <= 50) NOT NULL, -- Student Entrance Age
    entrance_year INT NOT NULL,                 -- Student Entrance Year
    class VARCHAR(20) NOT NULL                  -- Student Class
);

CREATE TABLE teachers(
    tid VARCHAR(5) NOT NULL PRIMARY KEY,        -- Teacher ID
    tname NVARCHAR(20) NOT NULL,                -- Teacher Name
    course NVARCHAR(20) NOT NULL                -- Teacher Course
);

CREATE TABLE courses (
    cid VARCHAR(7) NOT NULL PRIMARY KEY,        -- Course ID
    cname NVARCHAR(20) NOT NULL,                -- Course Name
    tid VARCHAR(5) FOREIGN KEY REFERENCES teachers(tid) ON DELETE SET NULL, -- Teacher ID
    credit DECIMAL(4,2) NOT NULL,               -- Course Credit
    grade VARCHAR(20) NOT NULL,                 -- Course Grade
    cancel_year INT                             -- Course Cancel Year
);

CREATE TABLE choose (
    sid VARCHAR(10) NOT NULL FOREIGN KEY REFERENCES students(sid) ON DELETE CASCADE, -- Student ID
    cid VARCHAR(7) FOREIGN KEY REFERENCES courses(cid) ON DELETE SET NULL,           -- Course ID
    tid VARCHAR(5) FOREIGN KEY REFERENCES teachers(tid) ON DELETE SET NULL,         -- Teacher ID
    score FLOAT,                              -- Score
    chosen_year INT NOT NULL                   -- Chosen Year
);
```

**Authorities' information table construction.** Three tables are created for each respective authority level: students, teachers, and administrators. Each of the separate tables contains usernames, as well as its associated password, which is a prerequisite before access to the program is granted to the user.

**Figure 6**

*Authorities Table Creation Commands*

```
CREATE TABLE account_students (
    username VARCHAR(15) NOT NULL,
    password VARCHAR(15) NOT NULL
);

CREATE TABLE account_teachers (
    username VARCHAR(15) NOT NULL,
    password VARCHAR(15) NOT NULL
);

CREATE TABLE account_admins (
    username VARCHAR(15) NOT NULL,
    password VARCHAR(15) NOT NULL
);
```

## Sample Data Insertion

Several sample data are inserted into the database for program testing and observation, which is done within the IDE with the SQL commands as the following, shown in figure 7.

**Figure 7**

*Table Insertion Queries for Students, Teachers, Courses, and Course Choosing*

```
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000001', 'Tom Scott', 'male', 19, 2020, 'CST2020');
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000002', 'Sheila Clives', 'female', 21, 2020, 'CST2020');
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000003', 'Mary Jane', 'female', 20, 2021, 'CST2021');
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000004', 'Thomas Zhang', 'male', 18, 2021, 'CST2021');
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000005', 'Jeffrey Ho', 'male', 19, 2021, 'CST2022');
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000006', 'Clay Moore', 'male', 23, 2021, 'CST2022');
INSERT INTO students
  (sid, sname, sex, entrance_age, entrance_year, class)
VALUES ('1000000007', 'Danny Flemmings', 'male', 18, 2021, 'CST2022');

INSERT INTO teachers
  (tid, tname, course)
VALUES ('T0001', 'Jefferson Logatto', 'Linear Algebra');
INSERT INTO teachers
  (tid, tname, course)
VALUES ('T0002', 'Matt Rockald', 'Intro to CS');
INSERT INTO teachers
  (tid, tname, course)
VALUES ('T0003', 'Miles Hughes', 'C++ Programming');
INSERT INTO teachers
  (tid, tname, course)
VALUES ('T0004', 'Scarlet Rosalia', 'Java Programming');
INSERT INTO teachers
  (tid, tname, course)
VALUES ('T0005', 'Ben Owen', 'Data Structures');
INSERT INTO teachers
  (tid, tname, course)
VALUES ('T0006', 'Richard Higgings', 'Algorithms');

INSERT INTO courses
  (cid, cname, tid, credit, grade)
VALUES ('C000001', 'Linear Algebra', 'T0001', 3.0, '1');
INSERT INTO courses
  (cid, cname, tid, credit, grade)
VALUES ('C000002', 'Intro to CS', 'T0002', 2.0, '1');
INSERT INTO courses
  (cid, cname, tid, credit, grade)
VALUES ('C000003', 'C++ Programming', 'T0003', 4.0, '1');
INSERT INTO courses
  (cid, cname, tid, credit, grade)
VALUES ('C000004', 'Java Programming', 'T0004', 4.0, '2');
INSERT INTO courses
  (cid, cname, tid, credit, grade)
VALUES ('C000005', 'Data Structures', 'T0005', 3.0, '3');
INSERT INTO courses
  (cid, cname, tid, credit, grade)
VALUES ('C000006', 'Algorithms', 'T0006', 3.5, '3');
```

```

INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000001', 'C000001', 'T0001', 98, 2020);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000001', 'C000002', 'T0002', 91, 2020);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000001', 'C000003', 'T0003', 93, 2020);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000001', 'C000004', 'T0004', 87.9, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000001', 'C000005', 'T0005', 92, 2022);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000001', 'C000006', 'T0006', 86.67, 2022);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000002', 'C000001', 'T0001', 92.1, 2020);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000002', 'C000002', 'T0002', 87.6, 2020);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000002', 'C000003', 'T0003', 77.9, 2020);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000002', 'C000004', 'T0004', 93, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000002', 'C000005', 'T0005', 88, 2022);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000002', 'C000006', 'T0006', 85.3, 2022);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000003', 'C000001', 'T0001', 77.9, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000003', 'C000002', 'T0002', 82.1, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000003', 'C000003', 'T0003', 73, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000003', 'C000004', 'T0004', 84, 2022);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000004', 'C000001', 'T0001', 83.6, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000004', 'C000002', 'T0002', 91.2, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000004', 'C000003', 'T0003', 70.1, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000004', 'C000004', 'T0004', 66.6, 2022);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000005', 'C000001', 'T0001', 91, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000005', 'C000002', 'T0002', 93.5, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000005', 'C000003', 'T0003', 81.2, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000006', 'C000001', 'T0001', 77.1, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000006', 'C000002', 'T0002', 90.1, 2021);
INSERT INTO choose
  (sid, cid, tid, score, chosen_year)
VALUES ('1000000006', 'C000003', 'T0003', 91.2, 2021);

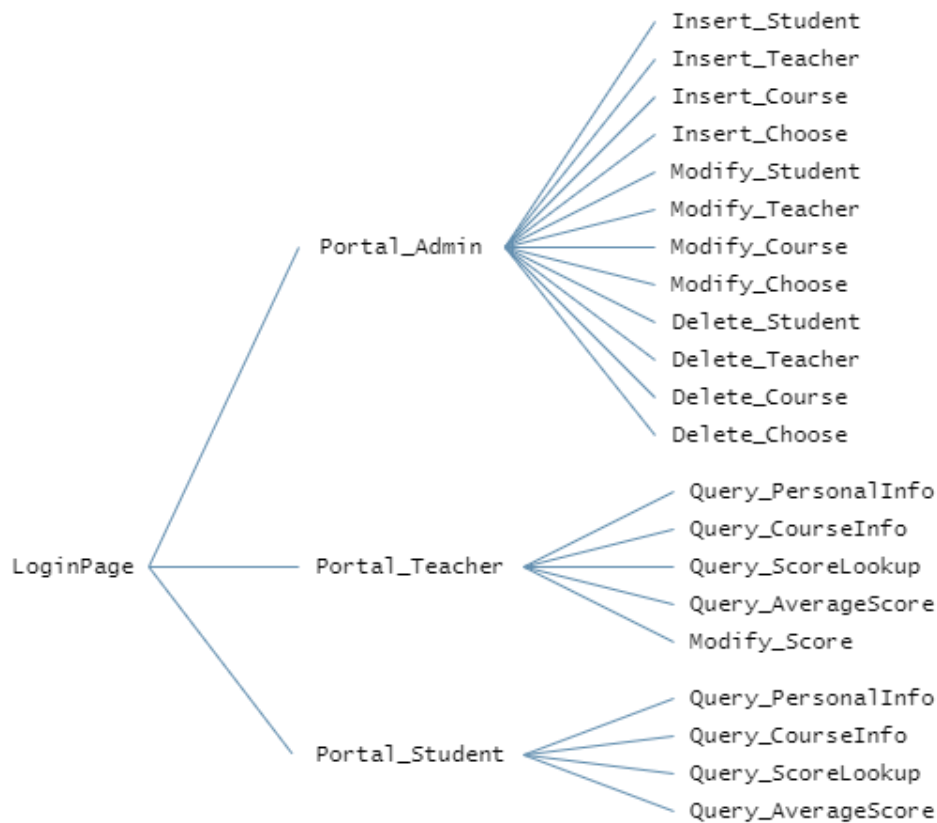
```



## UI Design with Windows Forms

As aforementioned, the UI components of the program are powered by Windows Forms (WinForms) .NET UI class library. With the utilities and tools provided with Visual Studio 2022, 21 UI pages are configured to support the program's complete functionality set, where relationships between each UI page is illustrated in figure 8.

**Figure 8**  
*UI Page Connections Diagram*

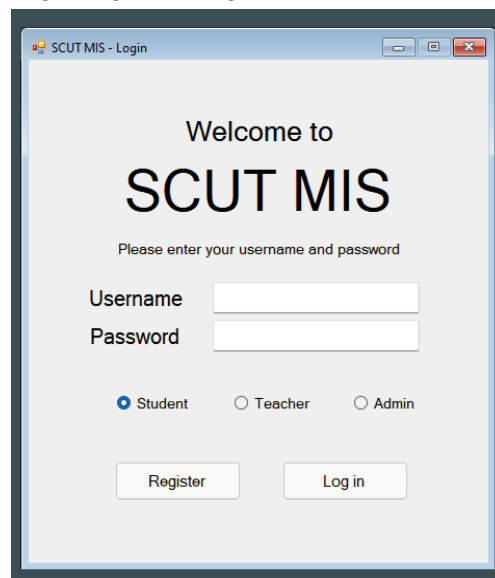


As shown in figure 8, the program is divided based on the authority levels determined during the login process, where administrators are granted the access to insert, modify, and delete entries of students, teachers, courses, or course choosing. Teachers and students share similar feature accessibilities, where they are permitted to query students' personal info, course info, view scores, as well as performing data lookup on average scores achieved.

Moreover, teachers are also given the additional privilege to perform alterations to the scores received by the students.

To access a certain authority level, users need to provide username and password, as well as selecting the account type from the radio button before registering or logging into an account. The username inputted in the textbox will be displayed in the portal display following it.

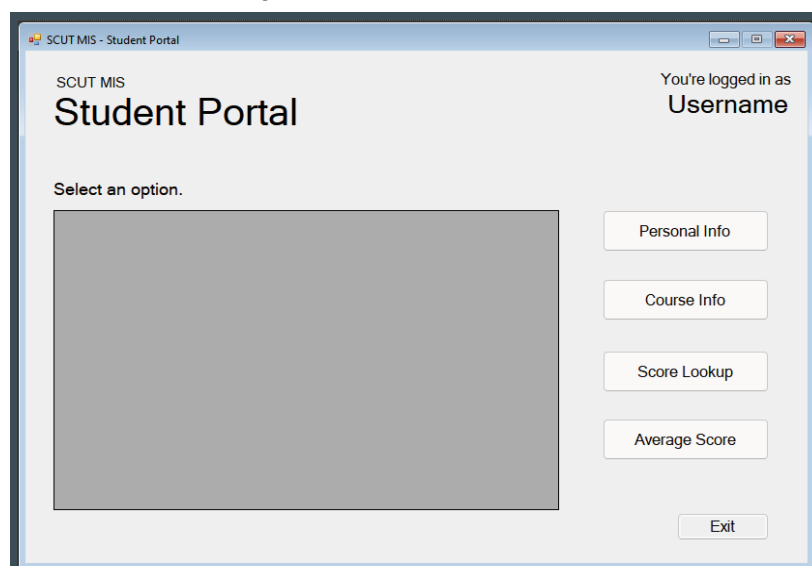
**Figure 9**  
*Login Page UI Design*



The image shows a window titled "SCUT MIS - Login". The main heading is "Welcome to SCUT MIS". Below it, a prompt says "Please enter your username and password". There are two text input fields: "Username" and "Password". Below these fields are three radio buttons labeled "Student", "Teacher", and "Admin", with "Student" selected. At the bottom, there are two buttons: "Register" and "Log in".

As illustrated by figure 10, after logging into the system as a student, the following student portal dialogue will be displayed onto the user.

**Figure 10**  
*Student Portal UI Design*

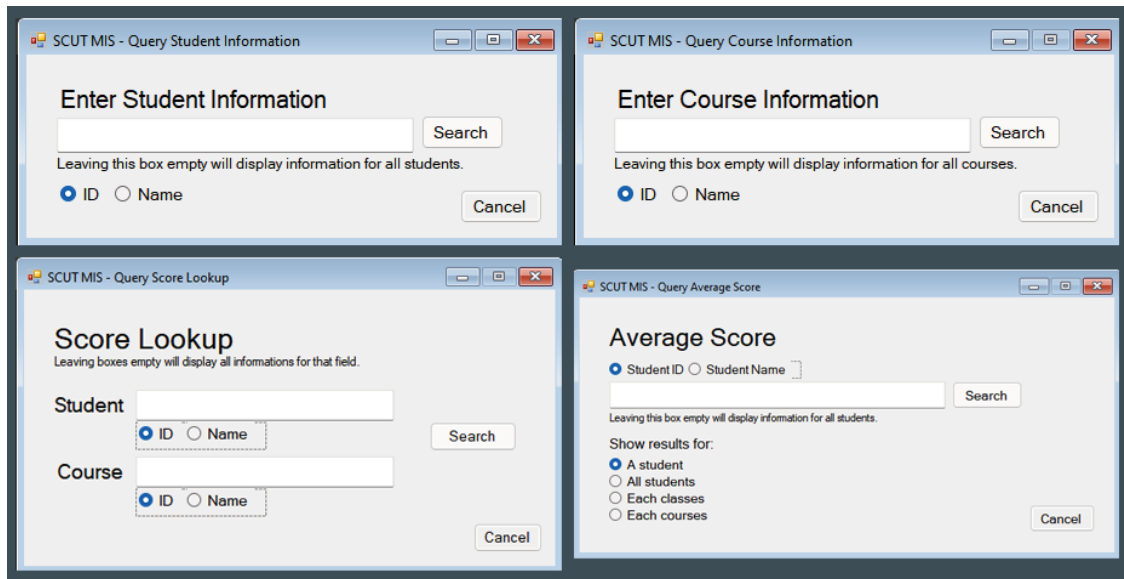


The image shows a window titled "SCUT MIS - Student Portal". The main heading is "SCUT MIS Student Portal". In the top right corner, it says "You're logged in as Username". Below the heading, there is a large gray rectangular area with the text "Select an option." above it. To the right of this area, there are four buttons stacked vertically: "Personal Info", "Course Info", "Score Lookup", and "Average Score". At the bottom right, there is an "Exit" button.

Each of the buttons from the student portal will generate the following pop-up dialogue respectively.

**Figure 11**

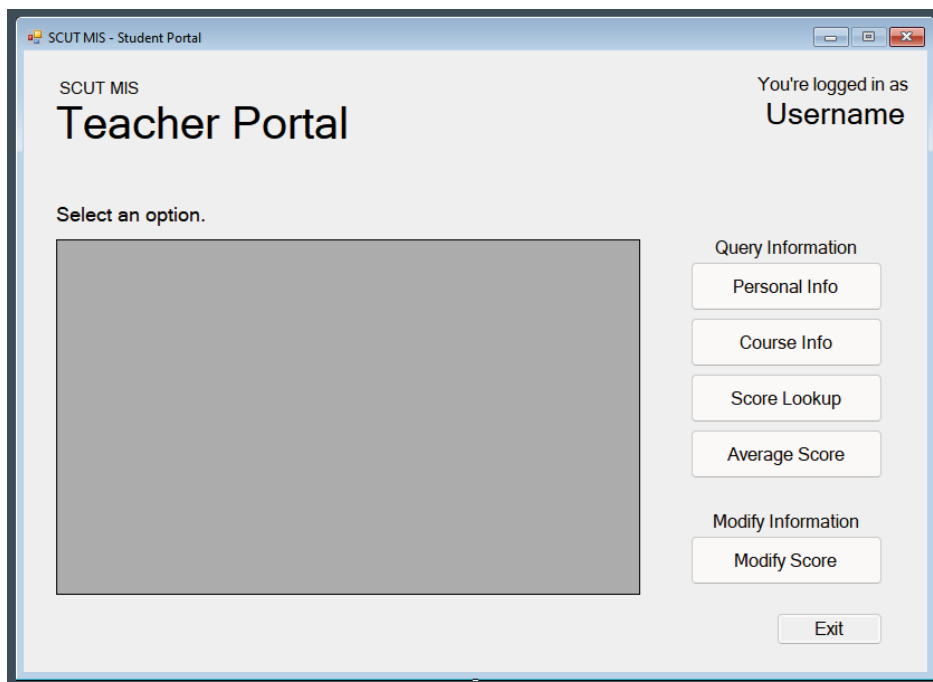
*UI Design for Student Portal's Popup Dialogues*



Logging in into the program with a teacher's account will display a teacher portal instead, whose UI is illustrated in figure 12.

**Figure 12**

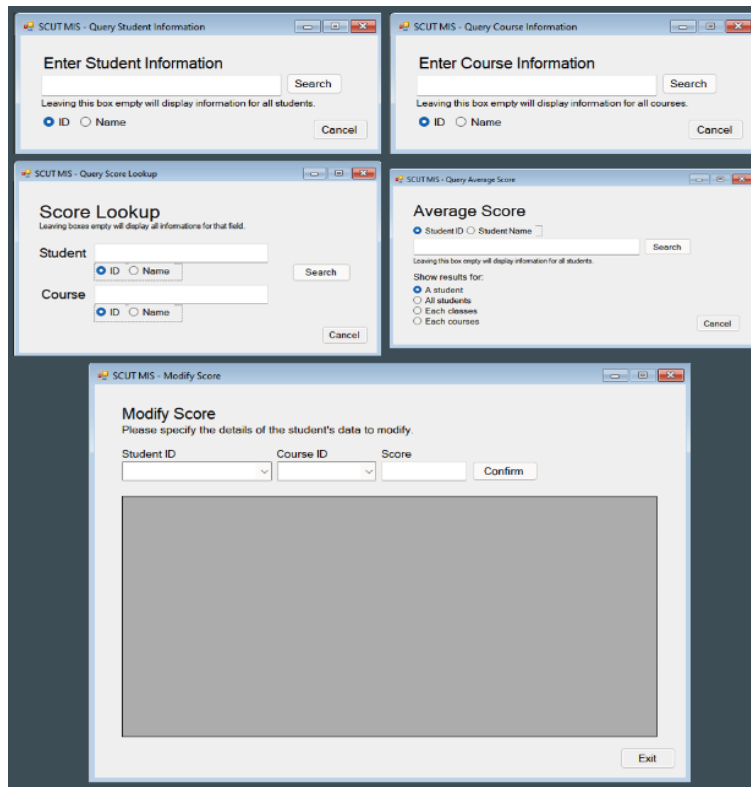
*Teacher Portal UI Design*



Similarly, the pop-up dialogues available from the teacher's portal is shown in the following.

**Figure 13**

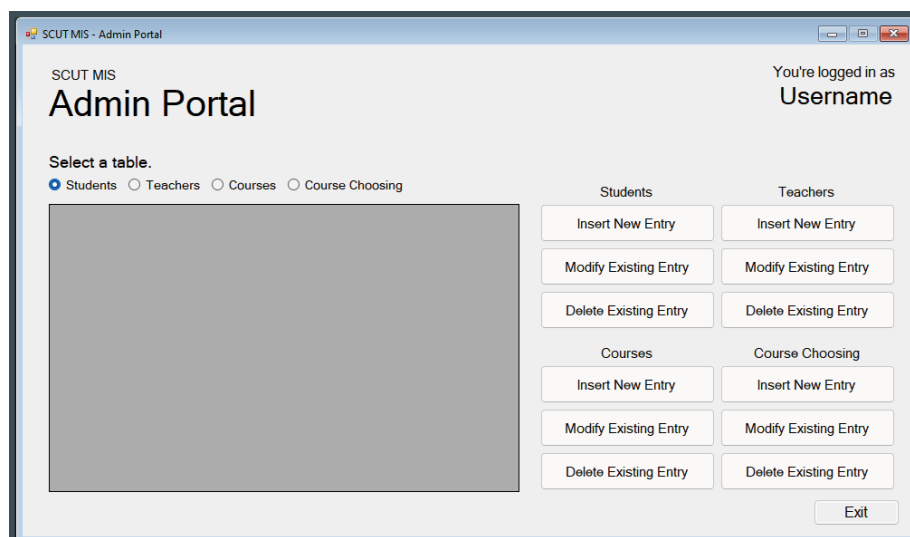
*UI Design for Teacher Portal's Popup Dialogues*



By selecting the administrator logging in method, the admin portal dialogue box will be displayed. Its UI design is shown in figure 14.

**Figure 14**

*Admin Portal UI Design*



Each of the buttons of the admin portal corresponds to individual pop-up dialogue boxes, shown in the following,

**Figure 15**

*UI Design for Admin Portal's Popup Dialogues*

The figure displays 12 individual pop-up dialogue boxes for the SCUT MIS admin portal, organized in a 4x3 grid. Each dialogue box has a title bar with the window name and standard OS controls. The dialogues are as follows:

- Row 1: Student Management**
  - SCUT MIS - Insert New Student:** "New Student Entry". Fields: ID, Name, Sex, Entrance Age, Entrance Year, Class. Buttons: Confirm, Exit.
  - SCUT MIS - Modify Student Info:** "Modify Student Info". Fields: Student ID (dropdown), Student Name, Sex (radio buttons: Male, Female), Entrance Age, Entrance Year, Class. Buttons: Save, Exit.
  - SCUT MIS - Delete Student Info:** "Delete Student Info". Fields: Student ID (dropdown), Student Name, Sex, Entrance Age, Entrance Year, Class. Buttons: Delete, Exit.
- Row 2: Teacher Management**
  - SCUT MIS - Insert New Teacher:** "New Teacher Entry". Fields: ID, Name, Course. Buttons: Confirm, Exit.
  - SCUT MIS - Modify Teacher Info:** "Modify Teacher Info". Fields: Teacher ID (dropdown), Teacher Name, Course. Buttons: Save, Exit.
  - SCUT MIS - Delete Teacher Info:** "Delete Teacher Info". Fields: Teacher ID (dropdown), Teacher Name, Course. Buttons: Delete, Exit.
- Row 3: Course Management**
  - SCUT MIS - Insert New Course:** "New Course Entry". Fields: Course ID, Course Name, Teacher ID (dropdown), Credit, Grade, Cancel Year. Buttons: Confirm, Exit.
  - SCUT MIS - Modify Course Info:** "Modify Course Info". Fields: Course ID (dropdown), Course Name, Teacher ID (dropdown), Credit, Grade, Cancel Year (Nullable). Buttons: Save, Exit.
  - SCUT MIS - Delete Course Info:** "Delete Course Info". Fields: Course ID (dropdown), Course Name, Teacher ID, Credit, Grade, Cancel Year (Nullable). Buttons: Delete, Exit.
- Row 4: Course Choosing**
  - SCUT MIS - Insert New Course:** "New Course Choosing Entry". Fields: Student ID (dropdown), Course ID (dropdown), Teacher ID (dropdown), Chosen Year. Buttons: Confirm, Exit.
  - SCUT MIS - Modify Course Choosing:** "Modify Course Choosing". Fields: Student ID (dropdown), Course ID (dropdown), Teacher ID (dropdown), Chosen Year. Buttons: Save, Exit.
  - SCUT MIS - Delete Course Choosing:** "Delete Course Choosing". Fields: Student ID (dropdown), Course ID (dropdown), Teacher ID (dropdown), Chosen Year. Buttons: Delete, Exit.

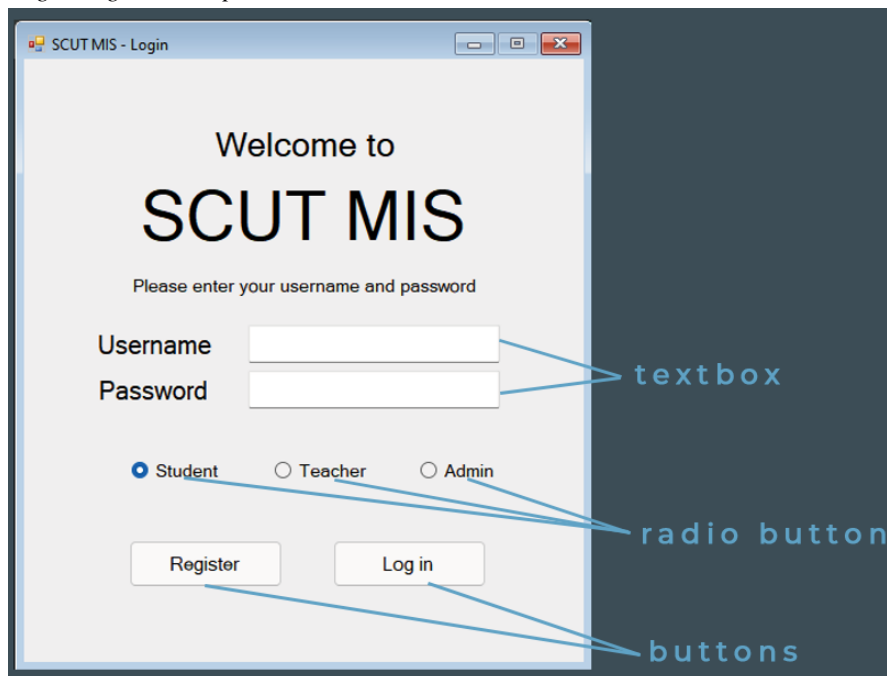
## Program Manual

### UI Components and Layout

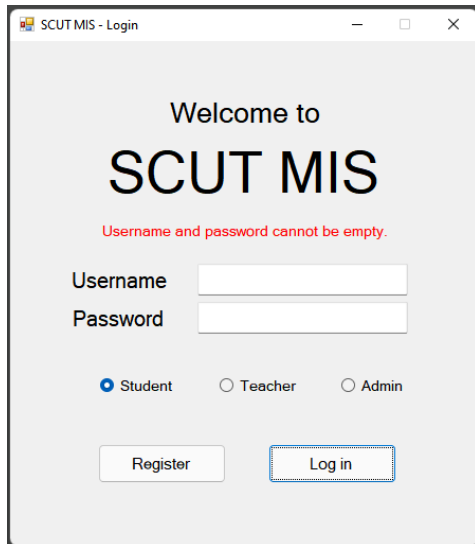
#### *Login Page*

The login page is comprised of two textboxes, two buttons, and three radio buttons. The two textboxes, `textbox_username` and `textbox_password`, are responsible for username and password input respectively. Meanwhile, the three radio buttons, `rbtn_student`, `rbtn_teacher`, and `rbtn_admin` are responsible for login method, whereas the two buttons, `btn_register` and `btn_login` are responsible for registering and logging users based on the username and passwords provided in the textboxes.

**Figure 16**  
*Login Page UI Components*



**Failsafe mechanisms.** As indicated in figure 17 and 18, the login page also provide failsafe mechanisms to handle errors, such as unregistered accounts, incorrect passwords, as well as other invalid credentials that users might provide during their login attempts. During such scenarios, the warning label will inform the user of the error cause, as well as changing colour to red.

**Figure 17***Empty username or password error*


SCUT MIS - Login

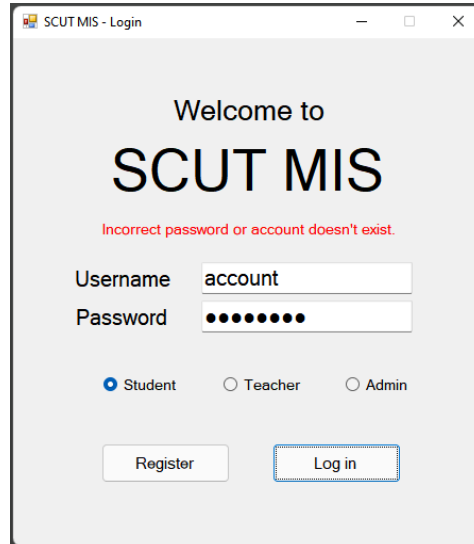
Welcome to  
**SCUT MIS**

Username and password cannot be empty.

Username

Password

☒ Student ☐ Teacher ☐ Admin

**Figure 18***Incorrect password or unregistered account error*


SCUT MIS - Login

Welcome to  
**SCUT MIS**

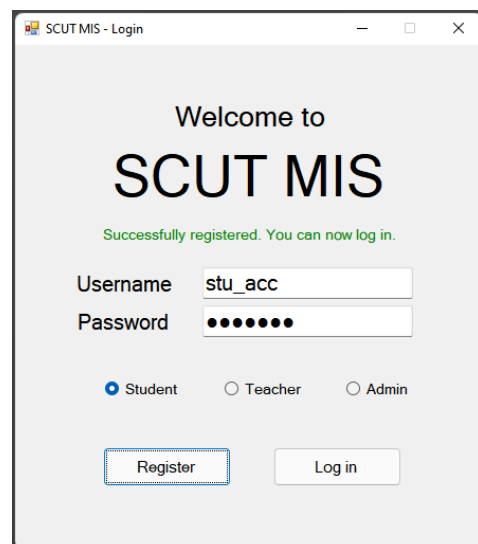
Incorrect password or account doesn't exist.

Username

Password

☒ Student ☐ Teacher ☐ Admin

**Successful actions.** When a registration or login attempt is successful, the warning label will turn to green, and notify users of the successful action.

**Figure 19***Successful registration message*


SCUT MIS - Login

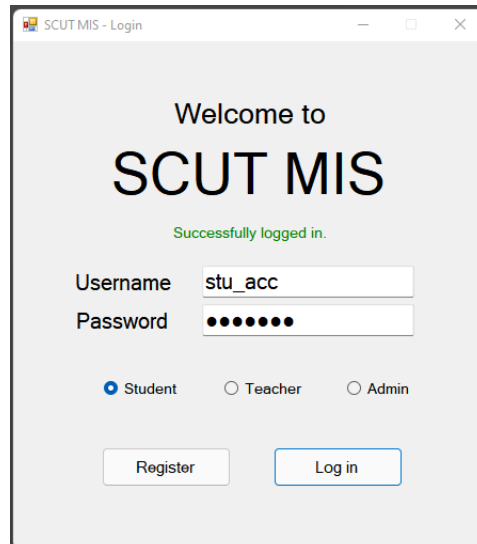
Welcome to  
**SCUT MIS**

Successfully registered. You can now log in.

Username

Password

☒ Student ☐ Teacher ☐ Admin

**Figure 20***Successful login message*


SCUT MIS - Login

Welcome to  
**SCUT MIS**

Successfully logged in.

Username

Password

☒ Student ☐ Teacher ☐ Admin

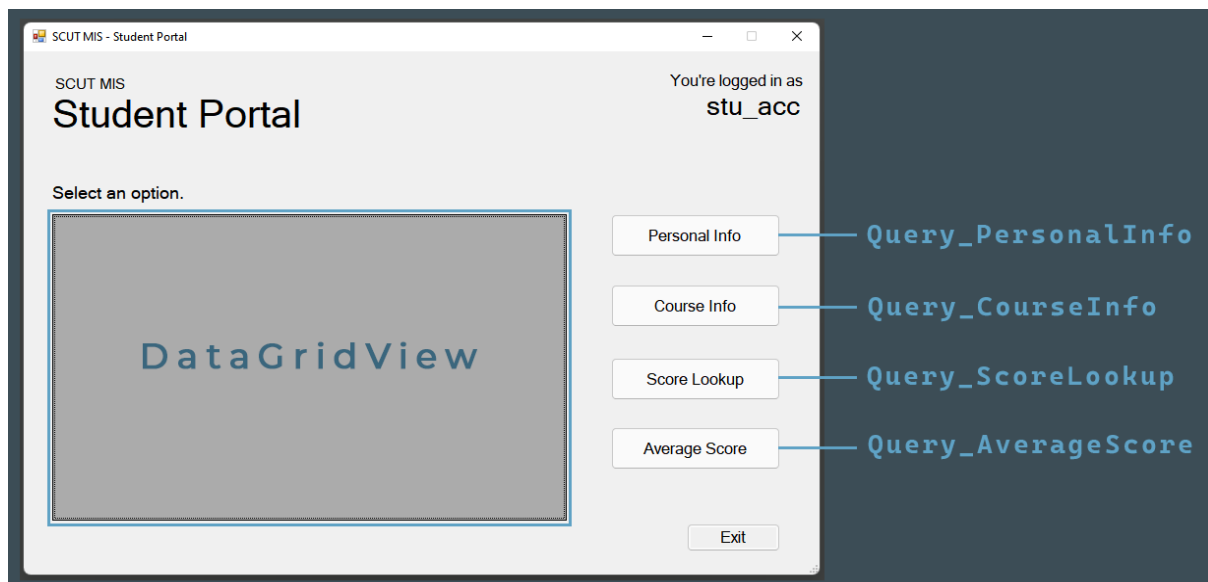
Upon successful login attempts, the program will display portals based on the login credentials, as well as the login method selected from the three radio buttons. The three portals are student portal, teacher portal, as well as admin portals.

### ***Student Portal***

The username used during the login process will be displayed by the label on the top right of the dialogue box. Moreover, the portal contains a `DataGridView` component which will display data tables queried by selecting any of the buttons on its right. Among the buttons, `btn_PersonalInfo` will generate a pop-up dialogue of `Query_PersonalInfo` WinForms page, while the remaining buttons, `btn_CourseInfo`, `btn_ScoreLookup`, and `btn_AverageScore` will generate `Query_CourseInfo`, `Query_ScoreLookup`, and `Query_AverageScore` respectively. Finally, when users wish to close the dialogue, they can click on the cross-shaped button on the top right of the window or click on the exit button provided on the bottom right of the dialogue box.

**Figure 21**

*Student Portal UI Components*

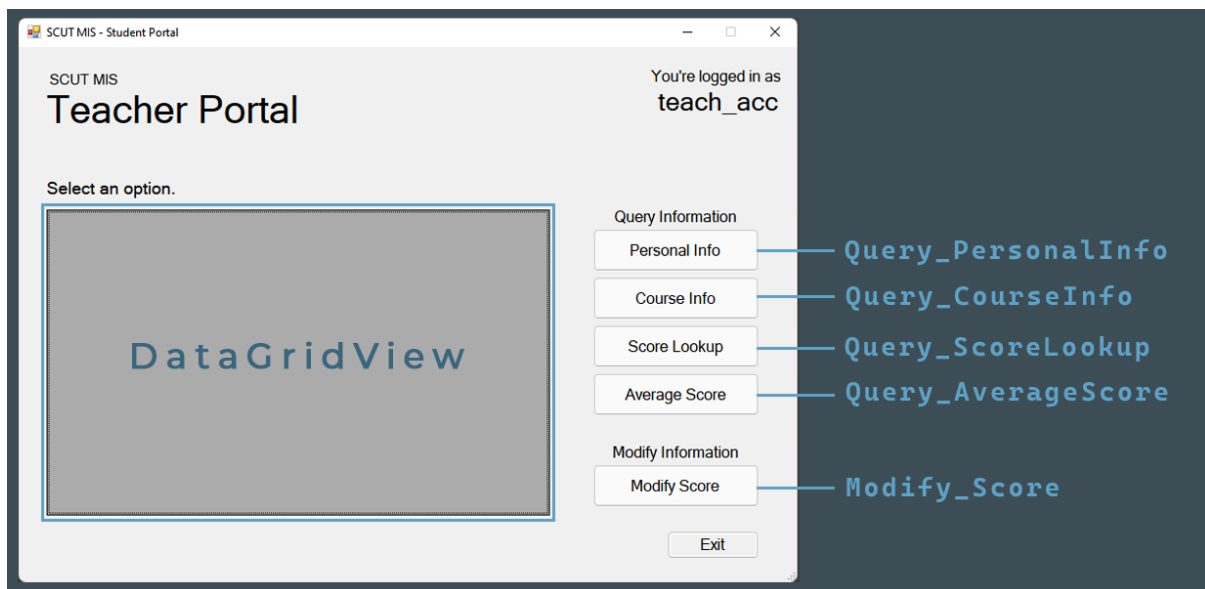


### ***Teacher Portal***

The teacher's portal is nearly identical to the student's portal. The difference lies in its fourth button, `btn_ModifyScore`, which will generate the pop-up dialogue `Modify_Score`, where teachers can alter scores assigned to students. Additionally, the buttons are also classified into two groups based on its utility, to query information or to modify information.



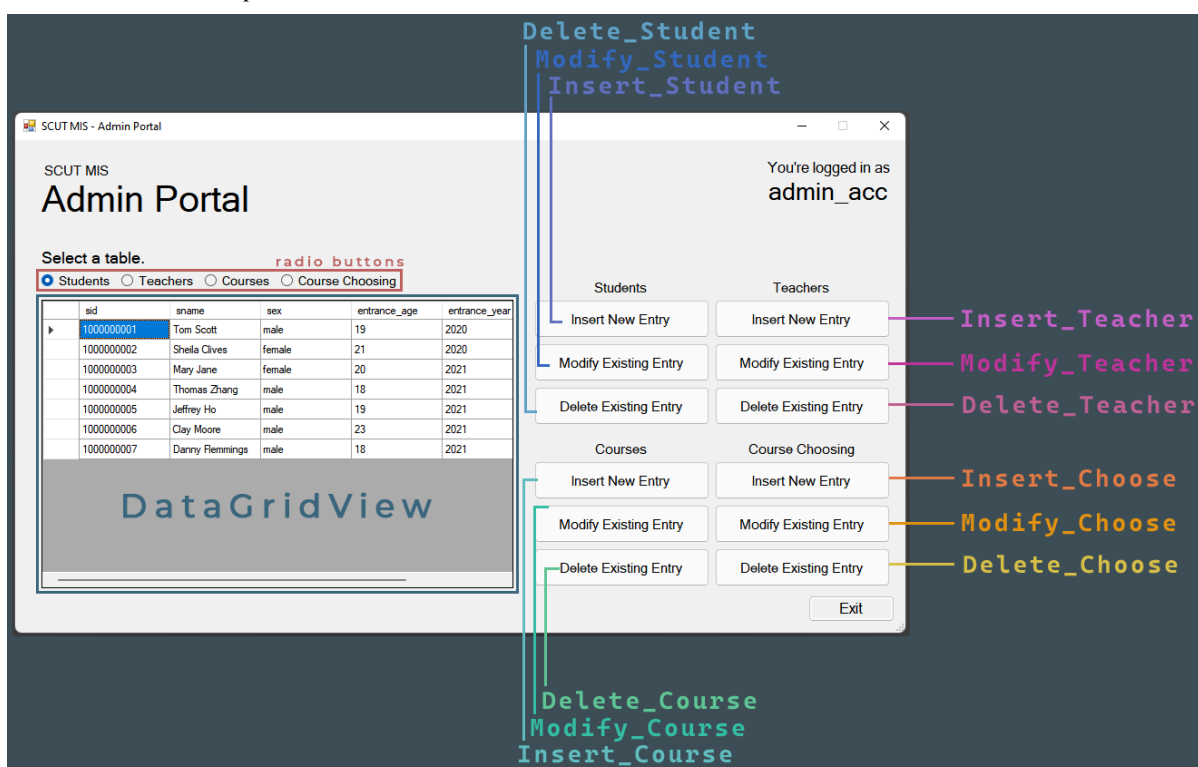
**Figure 21**  
Teacher Portal UI Components



### Admin Portal

Similar to other portals, the topmost section of the admin portal reveals general information of the window, as well as providing the username of the administrator account accessed.

**Figure 22**  
Admin Portal UI Components



The admin portal also has one `DataGridView` layout. However, the usage of the layout greatly differs from the portals prior, as it reveals all information of students, teachers, courses, and course choosing as determined by the selected radio buttons on top of it. Figure 23 reveals the inner workings of the feature, where clicking on any one of the four will issue a SQL command to retrieve all information of each table.

**Figure 23**

*Admin Portal Radio Button Source Code*

```
private void LoadDataGridView()
{
    string sqlQuery = "SELECT * FROM ";
    if (rbtn_Students.Checked)
        sqlQuery += "students";
    else if (rbtn_Teachers.Checked)
        sqlQuery += "teachers";
    else if (rbtn_Courses.Checked)
        sqlQuery += "courses";
    else if (rbtn_Choose.Checked)
        sqlQuery += "choose";
    else return;

    using (SqlConnection sqlConnection = new System.Data.SqlClient.SqlConnection(SqlHelper.CnnVal("database")))
    {
        using (SqlDataAdapter dataAdapter = new SqlDataAdapter(sqlQuery, sqlConnection))
        {
            using (SqlCommandBuilder sqlCommandBuilder = new SqlCommandBuilder(dataAdapter))
            {
                DataTable dataTable = new DataTable();
                dataAdapter.Fill(dataTable);
                dataGridView.DataSource = dataTable;
            }
        }
    }
}

4 references
private void rbtn_OnClick(object sender, EventArgs e) => LoadDataGridView();
```

Moreover, the buttons of the portal are grouped based on their target table. The buttons in students' group will generate the `Insert_Student`, `Modify_Student`, as well as `Delete_Student` dialogue boxes. The teachers' group will generate the `Insert_Teacher`, `Modify_Teacher`, and `Delete_Teacher` dialogue boxes, whereas the courses' group will generate the `Insert_Course`, `Modify_Course`, as well as `Delete_Course` dialogue boxes. The last group, course choosing's, will generate the `Insert_Choose`, `Modify_Choose`, and `Delete_Choose` dialogue boxes.

### ***Query\_PersonalInfo***

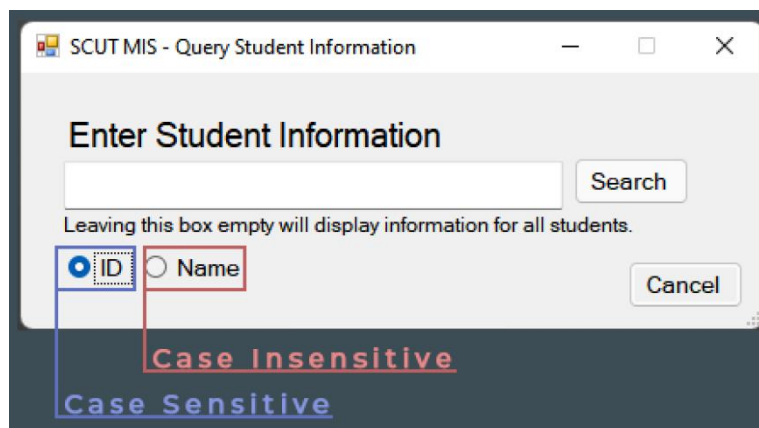
As aforementioned, this UI dialogue box is shown when a student's personal information is about to be queried. Figure 24 shows that there are two ways to provide

students' information identity, through either their student ID, or their names. These searching methods differ in the characters they receive. While the former is case-sensitive, the latter is case-insensitive.

Moreover, the warning label under the text box informs the user that if no information is provided, the system will query information for all students within the database.

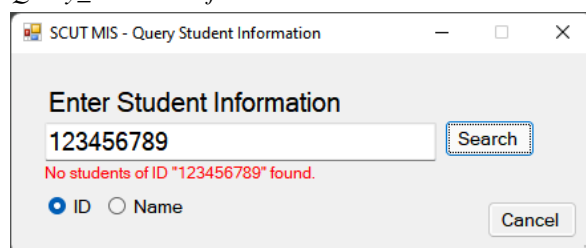
To finalize the query action, users need to click on the search button, or the cancel button to close the dialogue box.

**Figure 24**  
*Query\_PersonalInfo UI Components*

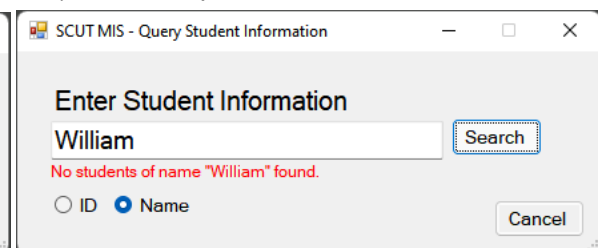


**Failsafe mechanisms.** In the event when the student ID or name does not match any of the records, the warning label will turn red and display the error message.

**Figure 25**  
*Query\_PersonalInfo ID Not Found Error*



**Figure 26**  
*Query\_PersonalInfo Name Not Found Error*

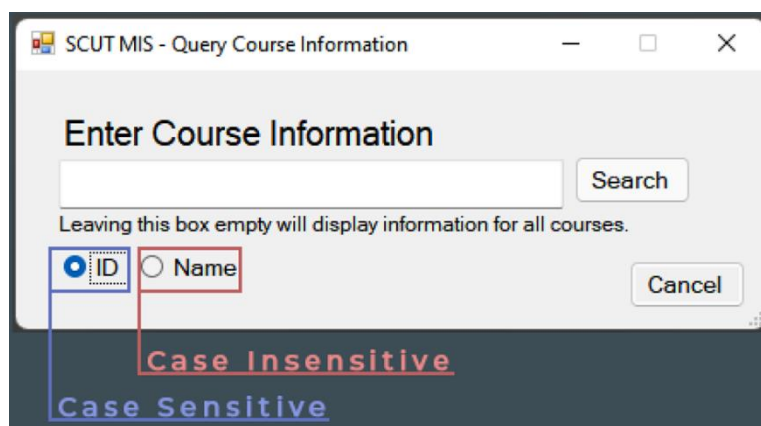


When no error is found in the information provided, the system will instead retrieve the requested data and close the pop-up box. The retrieved data will be displayed as a table in the DataGridView within the portal where the dialogue box was generated from.

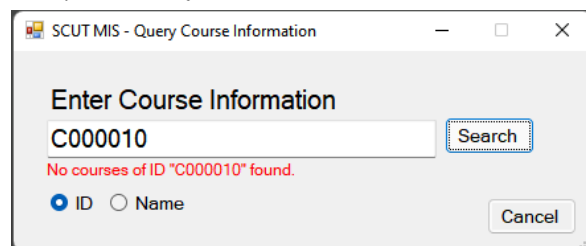
### *Query\_CourseInfo*

Similar to the previous dialogue box, *Query\_CourseInfo* receives two type of input, course id or course name, which are case-sensitive and case-insensitive respectively. Its warning label also informs the user that if no information is provided, all course information will be shown. Additionally, the failsafe mechanism, as well as its search and cancel button also work similarly to the dialogue box prior.

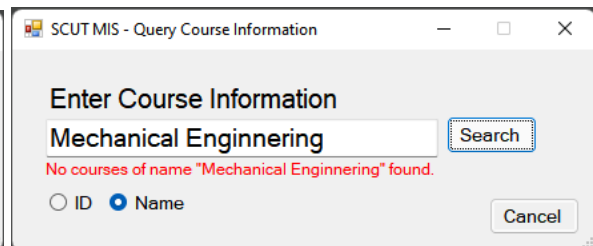
**Figure 27**  
*Query\_CourseInfo UI Components*



**Figure 28**  
*Query\_CourseInfo ID Not Found Error*



**Figure 29**  
*Query\_CourseInfo Name Not Found Error*



### *Query\_ScoreLookup*

To query the scores of individuals or courses, the id or name of the student and course needs to be provided. Similar to the previous dialogue boxes, the IDs are case-sensitive while the names are case-insensitive.

In the case where a box is left empty, all information of the field will be revealed. For instance, when student textbox is unfilled, all student's information will be revealed, and the same happens when the contents of the course textbox is blank.

**Figure 30***Query\_ScoreLookup UI Components*

SCUT MIS - Query Score Lookup

### Score Lookup

Leaving boxes empty will display all informations for that field.

Student

☒ ID ☐ Name

Course

☒ ID ☐ Name

Search

Cancel

### *Query\_AverageScore*

Users can retrieve the average scores of individual students, all students, students who share the same class, or students who share the same courses. These options are determined by selecting any of the four radio buttons on the bottom of the dialogue box.

**Individual student.** As figure 31 indicates, when the search option is set to “A student”, users can select to either input the ID or name student to be displayed in the query result. Leaving the textbox empty however, will display the average scores for all students.

**Figure 31***Query\_AverageScore UI Components (Individual Student)*

SCUT MIS - Query Average Score

### Average Score

Leaving this box empty will display information for all students.

☒ Student ID ☐ Student Name

Search

Show results for:

☒ A student

☐ All students

☐ Each classes

☐ Each courses

Cancel

**All students.** Seen on Figure 32, when the “All students” option is selected, the textbox and the radio buttons are disabled and the user is only left with three options, to change the search result settings, to continue with the query by clicking the search button, or to cancel and close the dialogue box instead.

**Figure 32**  
*Query\_AverageScore UI Components (All Students)*

**Students in the same class (each classes).** Figure 33 shows the display change when the option “Each classes” is selected instead. As no class ID is implemented within the database tables, the “ID” radio button is disabled while the “Class Name” is selected automatically. Moreover, the warning label under the textbox changed to inform the user of the outcome if it were to be left empty.

**Figure 33**  
*Query\_AverageScore UI Components (Each Classes)*

**Students in the same course.** As shown in figure 34, when this option is selected, the radio button's texts changed into "Course ID" and "Course Name" respectively, while the warning label informs the user that emptying the textbox will result in all courses being display instead.

**Figure 34**

*Query\_AverageScore UI Components (Each Courses)*

SCUT MIS - Query Average Score

### Average Score

☒ Course ID ☐ Course Name

Leaving this box empty will display information for all courses.

Show results for:

☐ A student  
☐ All students  
☐ Each classes  
☒ Each courses

### *Modify\_Score*

The UI dialogue is restricted to be only accessible to teachers. It comprises of two dropdown combo boxes, one textbox, and one `DataGridView` to display the students' ID, their course choosing, as well as scores obtained each.

**Figure 35**

*Modify\_Score UI Components*

SCUT MIS - Modify Score

### Modify Score

Please specify the details of the student's data to modify.

Student ID  Course ID  Score

Student ID	Student Name	Course Name	Course ID	Score
1000000001	Tom Scott	Linear Algebra	C000001	98
1000000001	Tom Scott	Intro to CS	C000002	91
1000000001	Tom Scott	C++ Programming	C000003	93
1000000001	Tom Scott	Java Programming	C000004	87.9
1000000001	Tom Scott	Data Structures	C000005	92
1000000001	Tom Scott	Algorithms	C000006	86.67
1000000002	Sheila Clives	Linear Algebra	C000001	92.1
1000000002	Sheila Clives	Intro to CS	C000002	87.6
1000000002	Sheila Clives	C++ Programming	C000003	77.9
1000000002	Sheila Clives	Java Programming	C000004	93
1000000002	Sheila Clives	Data Structures	C000005	88
1000000002	Sheila Clives	Algorithms	C000006	85.3
1000000003	Mary Jane	Linear Algebra	C000001	77.9
1000000003	Mary Jane	Intro to CS	C000002	82.1

As seen on figure 35, initially, only the student ID combo box is enabled on default. The remaining boxes for course ID and score, however, is only enabled after one of the available student ID is selected.

**Figure 36**

*Modify\_Score UI Components (Student ID Selection Result)*

The figure consists of two side-by-side screenshots of the 'SCUT MIS - Modify Score' application window. Both windows have the title 'SCUT MIS - Modify Score' and a subtitle 'Please specify the details of the student's data to modify.'.

The left screenshot shows the initial state. The 'Student ID' dropdown menu is open, displaying a list of student IDs: 1000000001, 1000000002, 1000000003, 1000000004, 1000000005, 1000000006, and 1000000007. The 'Course ID' and 'Score' fields are disabled. A table below the fields displays a list of students and their scores.

The right screenshot shows the state after a student ID has been selected. The 'Student ID' dropdown now displays '1000000001'. The 'Course ID' and 'Score' fields are now enabled. The table below the fields displays records for the selected student, Tom Scott, across various courses and their scores.

Student ID	Student Name	Course Name	Course ID	Score
1000000001	Tom Scott	Linear Algebra	C000001	98
1000000001	Tom Scott	Intro to CS	C000002	91
1000000001	Tom Scott	C++ Programming	C000003	93
1000000001	Tom Scott	Java Programming	C000004	87.9
1000000001	Tom Scott	Data Structures	C000005	92
1000000001	Tom Scott	Algorithms	C000006	86.67

Following the selection of a student ID, as shown in figure 36, the table displayed is filtered to only display the course choosing of the student ID selected, and teachers can modify the scores of the student by selecting a course ID of their course choosing to modify, followed by clicking on the confirmation button to execute the SQL query.

**Failsafe Mechanism.** In the event that there exists a student without any course choosing, the course id field is emptied and score box is set to disabled while the DataGridView display will show no records.

**Figure 37**

*Modify\_Score Failsafe Mechanism (Student with No Course Choosing)*

The screenshot shows the 'SCUT MIS - Modify Score' application window. The 'Student ID' dropdown menu is set to '1000000007'. The 'Course ID' and 'Score' fields are disabled. The table below the fields is empty, indicating that no records are displayed for this student ID.



### ***Insert\_Student***

The admin-accessible UI dialogue of **Insert\_Student** is comprised of 6 textboxes, where each act as a container for individual student information component, namely student ID, name, sex, entrance age, entrance year, and class respectively.

**Figure 38**  
*Insert\_Student UI Components*

**Parameters restriction.** Several restrictions are imposed to the textboxes, where student ID can be no longer than 10 characters, with the student's name consisting of no more than 20 characters. Moreover, the sex of the student can only be either “male” or “female”, while their age during entrance should be between 10 to 50 years of age and the class name of the student should be no longer than 20 characters. In scenarios where any of these criteria are not met, the system will display an error message for admins to revise their inputs.

**Figure 39**  
*Insert\_Student UI Error Messages*

The figure shows three instances of the 'New Student Entry' dialog box, each with a different validation error message in red text at the top:

- Left:** "Entrance age can only be between 10 and 50." The Entrance Age field contains the value '8'.
- Middle:** "Invalid student entrance year." The Entrance Year field contains the value '19 BCE'.
- Right:** "Student class exceeded character limit. (max 20)". The Class field contains the value 'ComputerScienceAndTechnol'.

Each dialog box includes fields for ID (10000001), Name (Cranel Rodriguez), Sex (male), Entrance Age, Entrance Year, and Class. It also has 'Confirm' and 'Exit' buttons at the bottom.

### ***Insert\_Teacher***

The admin-accessible UI dialogue is comprised of three textboxes, representing teacher's ID, name, and the course they taught.

**Figure 40**  
*Insert\_Teacher UI Components*

The figure shows the 'New Teacher Entry' dialog box. It has a title bar 'SCUT MIS - Insert New Teacher'. The main title is 'New Teacher Entry'. Below it is the instruction 'Please enter teacher details.' followed by three input fields: 'ID', 'Name', and 'Course'. At the bottom are 'Confirm' and 'Exit' buttons.

Moreover, restrictions are set to ensure the fields are inputted appropriately, where their teachers' ID should be a string of characters of no more than 5 with name and course taught consisting of no more than 20 characters.

**Figure 41**  
*Insert\_Teacher UI Error Messages*

The figure shows three instances of the 'New Teacher Entry' dialog box, each with a different validation error message in red text at the top:

- Left:** "Teacher ID exceeded character limit. (max 5)". The ID field contains the value 'T000000001'.
- Middle:** "Teacher name exceeded character limit. (max 20)". The Name field contains the value 'Kleinn Browningham Smithy'.
- Right:** "Course exceeded character limit. (max 20)". The Course field contains the value 'Introduction to Computer Sci'.

Each dialog box includes fields for ID, Name, and Course. It also has 'Confirm' and 'Exit' buttons at the bottom.

## *Insert\_Course*

The admin-accessible UI dialogue is comprised of five textboxes and one combo box dropdown, representing course ID, name, teacher ID, course credit, course grade and the cancellation year of the course.

**Figure 42**  
*Insert\_Course UI Components*

Following the listed requirements of the projects, restrictions are imposed within the dialogue. Notably, course ID should be comprised of no longer than 7 characters, while course name and grade has the character limit of 20. The teacher in charge of the course should be listed as well, by providing the ID of an existing teacher. During inputting the data, admins need to comply these requirements. Should any data be invalid, error will be shown.

**Figure 43**  
*Insert\_Course Error Messages*

The figure consists of three side-by-side screenshots of the "SCUT MIS - Insert New Course" window. Each screenshot shows a form titled "New Course Entry" with fields for Course ID, Course Name, Teacher ID, Credit, Grade, and Cancel Year. The first screenshot displays the error message "Invalid credit. (expected a positive decimal ##.##)" because the Credit field contains "666.88". The second screenshot displays the error message "Course grade exceeded character limit. (max.20)" because the Grade field contains a long string of zeros. The third screenshot displays the error message "Invalid cancel year." because the Cancel Year field contains "14BCE". In each case, the "Confirm" button is highlighted with a red dashed border.

*Insert\_Choose*

The admin-accessible UI dialogue is comprised of three combo box dropdowns, each representing ID for students, courses and teachers, with one textbox for inputting the course's chosen year.

**Figure 44**  
*Insert\_Choose UI Components*

SCUT MIS - Insert New Course

## New Course Choosing Entry

Please enter course details. (cancel year is nullable)

Student ID

Course ID

Teacher ID

Chosen Year

Confirm Exit

Restrictions are set for the IDs chosen or entered to only be from an existing entity from each respective field. Moreover, the chosen year need to be before the course's cancellation year and is checked prior to the course choosing's registration into the database.

**Figure 45**  
*Insert Choose UI Error Messages*

The figure displays four sequential screenshots of the 'New Course Choosing Entry' form, illustrating different validation states:

- Invalid Student ID:** The 'Student ID' field contains '1000000020', which is highlighted in red. The 'Course ID' is 'C000050', 'Teacher ID' is 'T0020', and 'Chosen Year' is '2025'.
- Invalid Course ID:** The 'Student ID' field contains '1000000001'. The 'Course ID' field contains 'C000050', which is highlighted in red. 'Teacher ID' is 'T0020' and 'Chosen Year' is '2025'.
- Invalid Teacher ID:** The 'Student ID' field contains '1000000001'. The 'Course ID' field contains 'C000001'. The 'Teacher ID' field contains 'T0020', which is highlighted in red. 'Chosen Year' is '2025'.
- Course was cancelled at 2022:** A red message 'Course was cancelled at 2022' is displayed at the top. The 'Student ID' field contains '1000000001', 'Course ID' is 'C000001', 'Teacher ID' is 'T0001', and 'Chosen Year' is '2025'.

Each form includes 'Confirm' and 'Exit' buttons at the bottom.

### ***Modify\_Student***

The admin-accessible UI dialogue is comprised of one combo box dropdown, where admins can choose a student ID to select one student for their data to be modified. The detail fields are comprised of textboxes and a radio button, where the student's name, entrance age, entrance year, as well as class will be contained, and their sex will be chosen.

Upon the selection of a student, the system will retrieve their personal data from the database, display them and enable the textboxes for possible alterations.

The restrictions set to the dialogue is implemented to be similar to **Insert\_Student**'s.

**Figure 46**  
*Modify\_Student UI Components*

The figure displays two instances of the 'Modify Student Info' dialog box. Both windows have a title bar that reads 'SCUT MIS - Modify Student Info'. The main heading is 'Modify Student Info' with a subtitle 'Select an existing student ID to modify.'.

The left window shows the initial state with empty fields:
 

- Student ID: A dropdown menu.
- Details:
  - Student Name: An empty text box.
  - Sex: Radio buttons for 'Male' and 'Female'.
  - Entrance Age: An empty text box.
  - Entrance Year: An empty text box.
  - Class: An empty text box.
- Buttons: 'Save' and 'Exit'.

The right window shows the state after data retrieval:
 

- Student ID: A dropdown menu showing '1000000001'.
- Details:
  - Student Name: A text box containing 'Tom Scott'.
  - Sex: Radio buttons for 'Male' (selected) and 'Female'.
  - Entrance Age: A text box containing '19'.
  - Entrance Year: A text box containing '2020'.
  - Class: A text box containing 'CST2020'.
- Buttons: 'Save' and 'Exit'.

### ***Modify\_Teacher***

The admin-accessible UI dialogue is comprised of one combo box dropdown, where the ID of the teachers can be selected and retrieved, while two textboxes are contained in the details section, where the teacher's name and the course they taught is listed.

Upon selecting a teacher's ID, the relevant information of the teacher will be retrieved and displayed into the editable textboxes.

The restrictions imposed within the dialogue is similarly implemented as **Insert\_Teacher**'s.

**Figure 47***Modify\_Teacher UI Components*

The figure shows two side-by-side screenshots of the 'Modify Teacher Info' dialog box. The left screenshot shows the 'Teacher ID' dropdown menu open. The right screenshot shows the 'Teacher ID' dropdown menu with 'T0001' selected. Both screenshots show the 'Details' section with 'Teacher Name' (Jefferson Logatto) and 'Course' (Linear Algebra) text boxes, and 'Save' and 'Exit' buttons.

### *Modify\_Course*

The admin-accessible UI dialogue is comprised of one combo box under the title, where the ID of the course can be selected, which will retrieve the relevant information of the course and allow the details section to be alterable, where the course name, teacher's ID, course credit, grade, and the course's cancellation year is listed.

Notably, the restrictions imposed within the dialogue is similarly implemented as *Insert\_Course*'s.

**Figure 48***Modify\_Course UI Components*

The figure shows two side-by-side screenshots of the 'Modify Course Info' dialog box. The left screenshot shows the 'Course ID' dropdown menu open. The right screenshot shows the 'Course ID' dropdown menu with 'C000001' selected. Both screenshots show the 'Details' section with 'Course Name' (Linear Algebra), 'Teacher ID' (T0001), 'Credit' (3.00), 'Grade' (1), and 'Cancel Year (Nullable)' text boxes, and 'Save' and 'Exit' buttons.

## ***Modify\_Choose***

The admin-accessible UI dialogue has three combo box dropdowns under its title, where the students', courses', and teachers' ID need to be provided before any modifications take place.

Its details section contains the information of the choosing's student ID, course ID, teacher ID, and its chosen year, which upon clicking the save button, the renewed info will be stored into the database.

**Figure 49**  
*Modify\_Choose UI Components*

The figure displays two instances of the 'Modify Course Choosing' dialog box. The left instance shows the initial state with empty dropdowns for Student ID, Course ID, and Teacher ID. The right instance shows the same dialog box with the dropdowns populated with values: Student ID '1000000001', Course ID 'C000001', and Teacher ID 'T0001'. The 'Details' section also shows populated fields for Student ID, Course ID, Teacher ID, and Chosen Year '2020'.

Aside from the similar restrictions to **Insert\_Choose** imposed onto the dialogue, in a scenario where the student ID, course ID, and teacher ID combination does not exist, the system will throw an error and inform the user of no such entry found within the database.

**Figure 50**  
*Modify\_Choose UI Error Message*

The figure shows the 'Modify Course Choosing' dialog box with an error message. The title bar says 'SCUT MIS - Modify Course Choosing'. The main title is 'Modify Course Choosing'. Below the title, a red error message reads 'Course choosing not found.' The dropdowns for Student ID, Course ID, and Teacher ID are populated with '1000000001', 'C000001', and 'T0003' respectively. The 'Details' section shows populated fields for Student ID, Course ID, Teacher ID, and Chosen Year '2020'.

### ***Delete\_Student***

The admin-accessible dialogue contains one combo box dropdown, where an ID of a student can be selected. By selecting one student, the relevant information of them will be displayed under the details section, and upon clicking the delete button, their records, along with their course choosing, will be deleted.

**Figure 51**

*Delete\_Student UI Components*

The figure shows two instances of the 'Delete Student Info' dialog box. The left instance has the 'Student ID' dropdown menu open. The right instance shows the details for a selected student: 'Tom Scott' (ID: 1000000001, Sex: male, Entrance Age: 19, Entrance Year: 2020, Class: CST2020). Both instances have 'Delete' and 'Exit' buttons at the bottom.

### ***Delete\_Teacher***

The admin-accessible dialogue contains one combo box dropdown, where the ID of the teacher will need to be provided to correct identify the teacher entry to be deleted. Upon selection, the details of the teachers will be revealed, and the action can be completed by clicking the delete button.

**Figure 52**

*Delete\_Teacher UI Components*

The figure shows two instances of the 'Delete Teacher Info' dialog box. The left instance has the 'Teacher ID' dropdown menu open. The right instance shows the details for a selected teacher: 'Jefferson Logatto' (ID: T0001, Course: Linear Algebra). Both instances have 'Delete' and 'Exit' buttons at the bottom.



### ***Delete\_Choose***

The admin-accessible dialogue box contains three combo box dropdowns to identify the course choosing entry to delete. Once valid student ID, course ID, and teacher ID are correctly identified, clicking the search button will verify the input and retrieve the corresponding information onto the display.

**Figure 53**

*Delete\_Choose UI Components*

The figure displays two side-by-side screenshots of the 'Delete Course Choosing' dialog box in the SCUT MIS system. Both windows have the title 'SCUT MIS - Delete Course Choosing' and a subtitle 'Delete Course Choosing' with the instruction 'Select an existing entry to delete.'

**Left Screenshot (Initial State):**

- Three dropdown menus are present: 'Student ID', 'Course ID', and 'Teacher ID'. All are currently empty.
- A 'Search' button is located to the right of the dropdowns.
- The 'Details:' section lists the following fields: 'Student ID', 'Course ID', 'Teacher ID', and 'Chosen Year'. The values displayed are 'Student', 'Course', 'Teacher', and 'Chosen Year' respectively.
- At the bottom, there are 'Delete' and 'Exit' buttons.

**Right Screenshot (After Search):**

- The 'Student ID' dropdown now contains '1000000001'.
- The 'Course ID' dropdown now contains 'C000001'.
- The 'Teacher ID' dropdown now contains 'T0001'.
- The 'Search' button is highlighted with a blue border.
- The 'Details:' section now displays the retrieved information: 'Student ID 1000000001', 'Course ID C000001', 'Teacher ID T0001', and 'Chosen Year 2020'.
- The 'Delete' and 'Exit' buttons remain at the bottom.

## **Conclusion**

All in all, the design of the project has followed the general system design procedures, whilst satisfying the project requirements specified to be implemented, notably through SQL queries, managing students', teachers', courses', and course choosings' information, and to create a management information system with varying access privilege permission levels for students, teachers, and administrators, achieved with the C# programming language, Windows Forms GUI, and Microsoft SQL Server.

Covering all user input or misinput, the project is also designed with all possible scenarios in mind, by introducing failsafe mechanisms and imposing restrictions on data entry, modification, and deletion.

Ultimately, the project has been successfully in designing and implementing all requirements specified, in which is built with real-world scenario in mind.