# Project Description: Student Grading System

## Team Members

- Saleh Zohir Alomair (431109449)

- Khalifah Suliman Alkhalifah (431109692)

- Faisal Abdullah Alnasyan (431107944)

- Azzam Mohammed Alwahbi (431107748)

### 1. Application Overview

The Student Grading System is a comprehensive software application designed to streamline the management of student data, grades, and notifications. The system provides separate interfaces for administrators, teachers, and students, offering functionalities tailored to their specific roles.

## Key Features

### 1. Login and Registration System

- A user-friendly GUI for login and registration.

- Role-based access (Student, Teacher, Admin).

### 2. Admin Dashboard

- Manage users, courses, and notifications.

- Add, edit, and delete user accounts, courses, grades, and notifications.

### 3. Teacher Dashboard

- View and manage student grades.

- Add notifications for students.

### 4. Student Dashboard

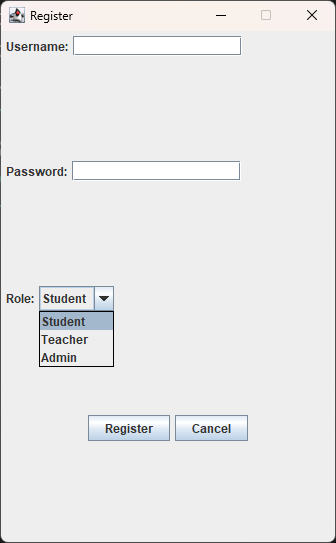
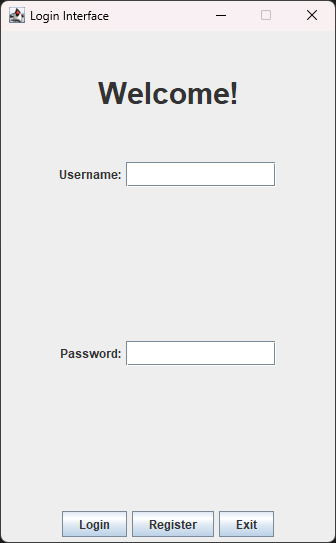
- View personal report, grades and notifications.

### 5. Database Management

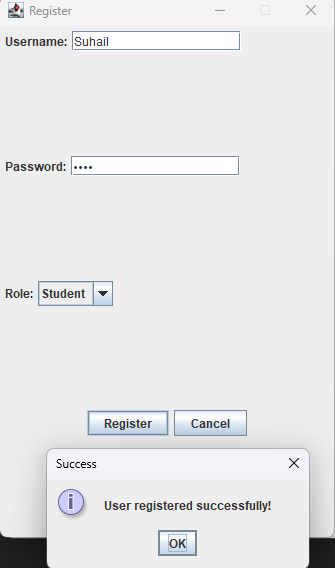
- Secure storage and retrieval of user data, grades, and notifications.

- MySQL database integration for dynamic data handling.

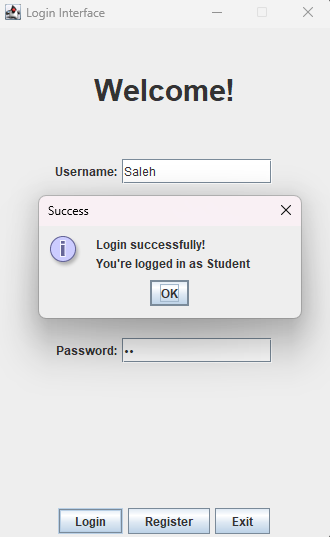
### 2. Gui Elements

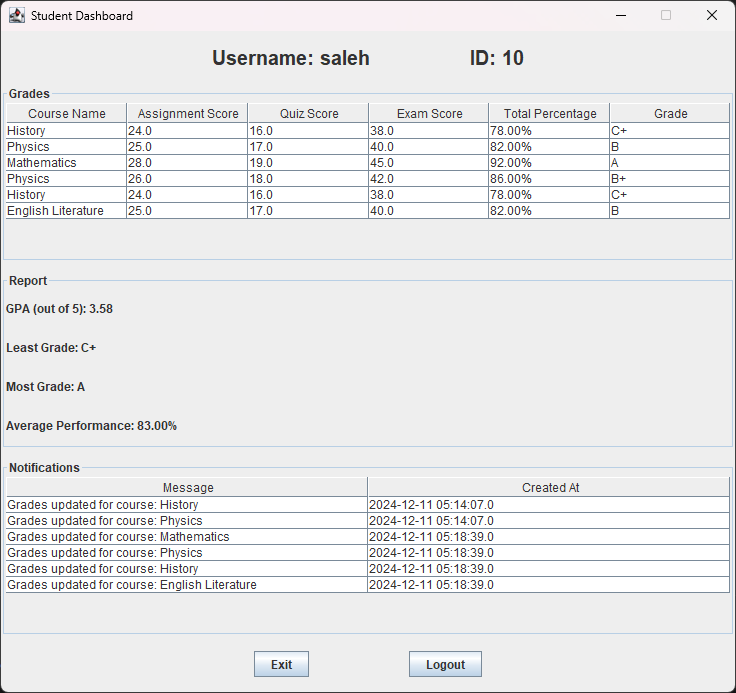


### Registering a new user

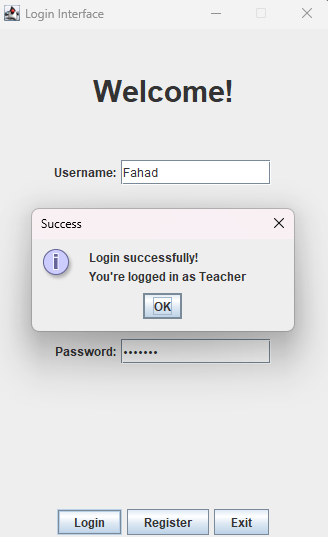


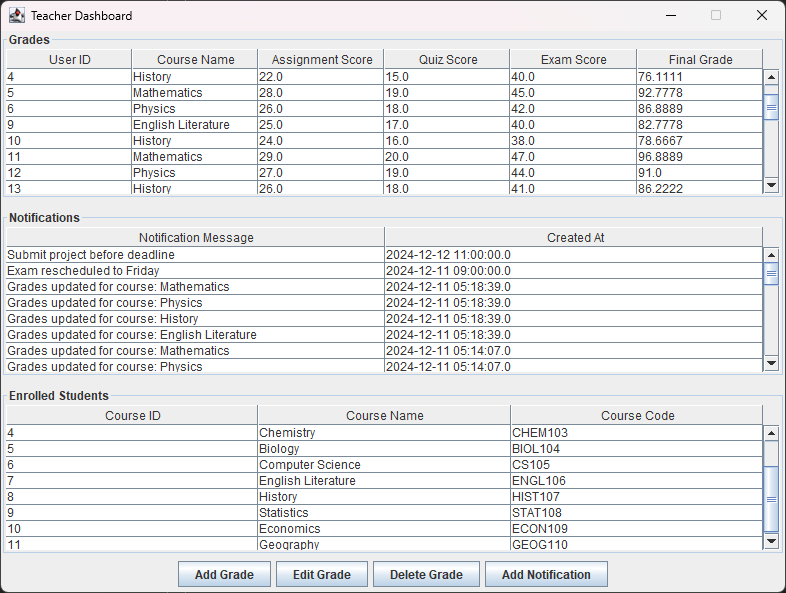
### When the logged user is Student:



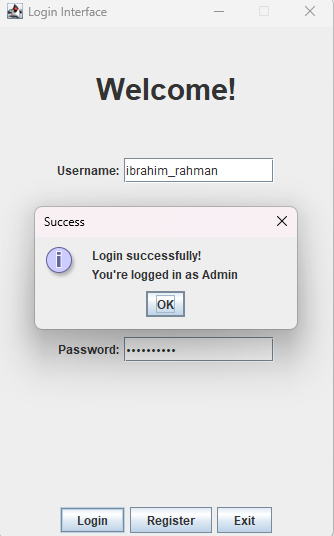


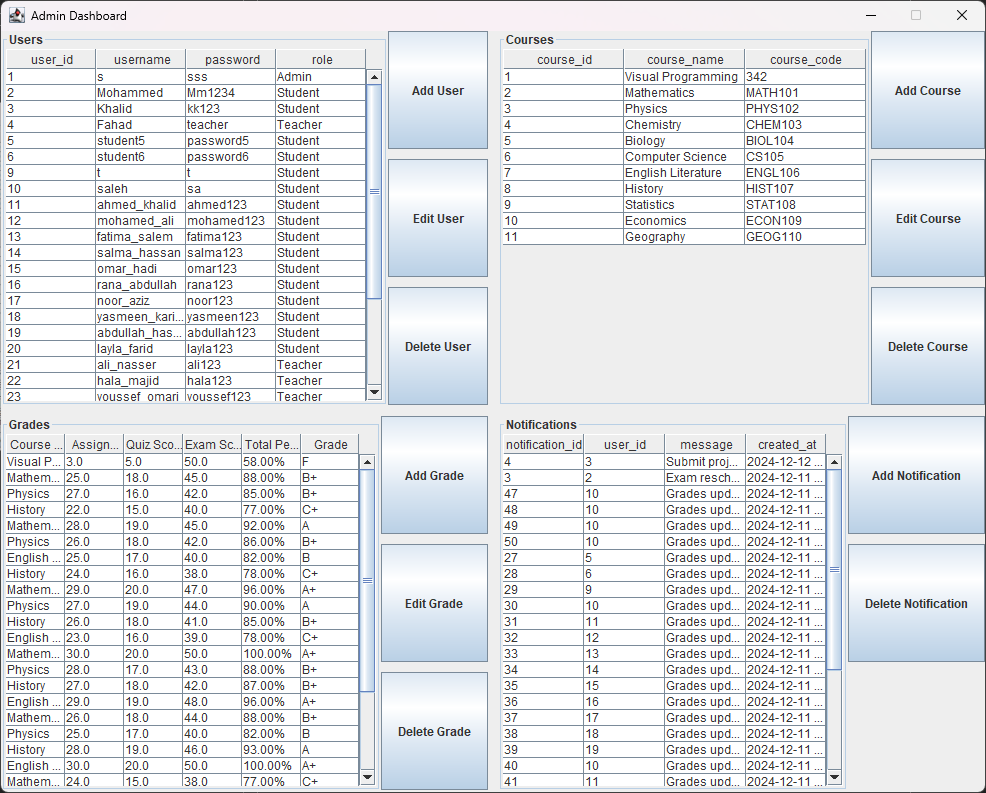
### When the logged user is Teacher:





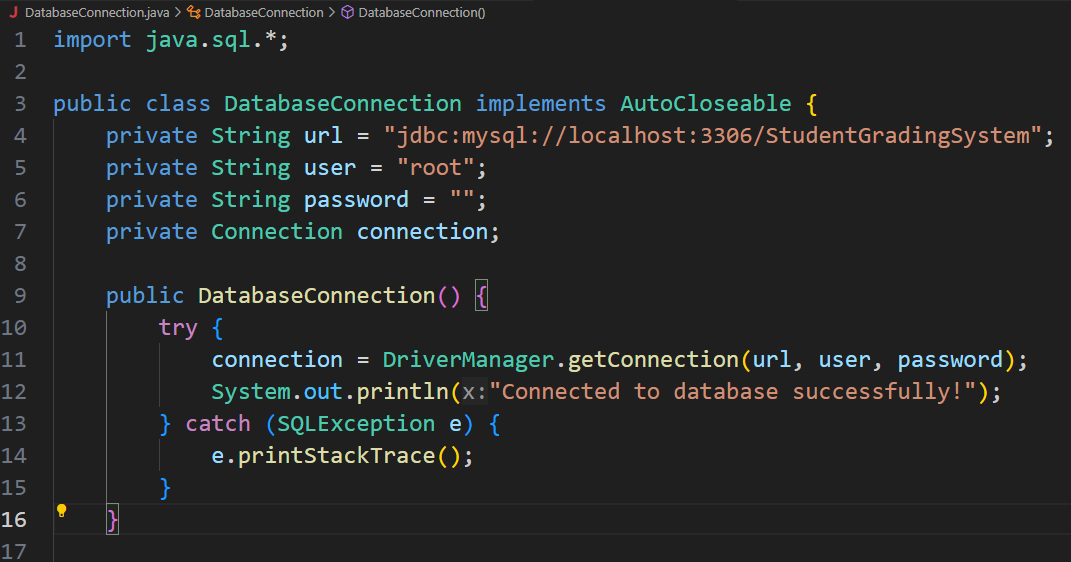
### When the logged user is Admin:





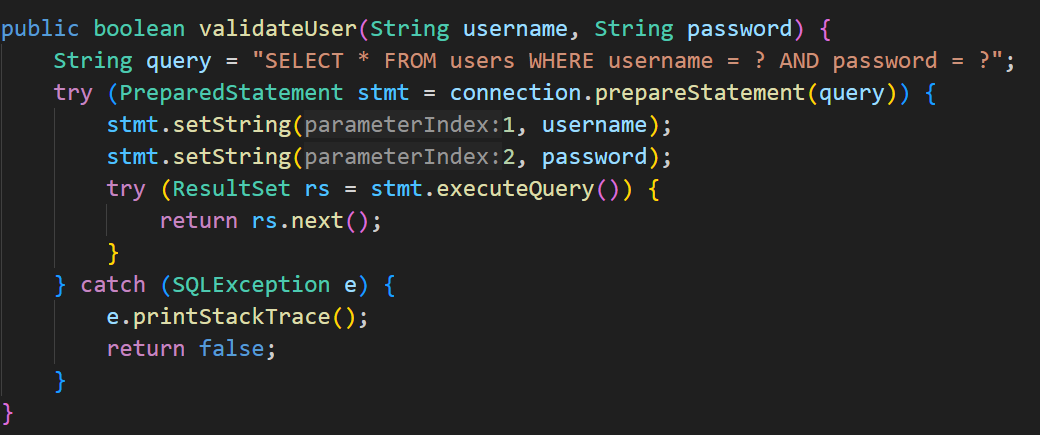
### 3. Database Connection

It has a whole class which connects the project to my phpMyAdmin localhost database in the start of the class.



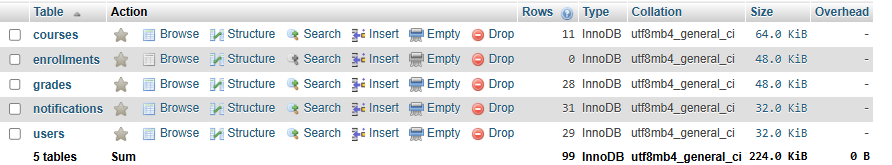
All the database methods are on this class.

For example, The method that takes the User Username input and the User Password input in the LoginGui and checks if it validate from the Users table on the database



### phpMyAdmin localhost:

It has multiple tables and each one of them contains the required columns

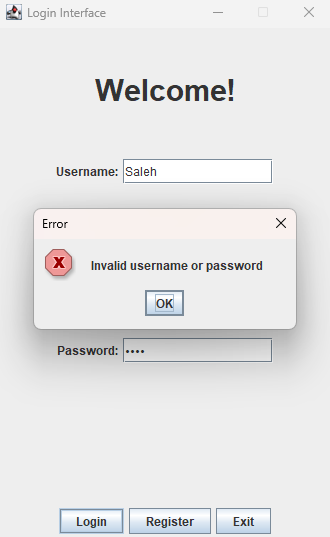


### 4. Exception Handling

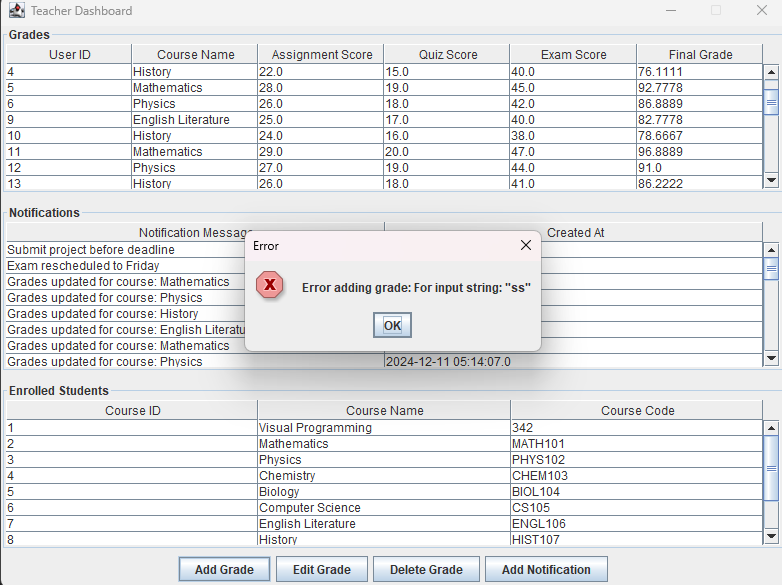
It covers a very amount of exception handling, almost every exception that may happen.

Some of them:

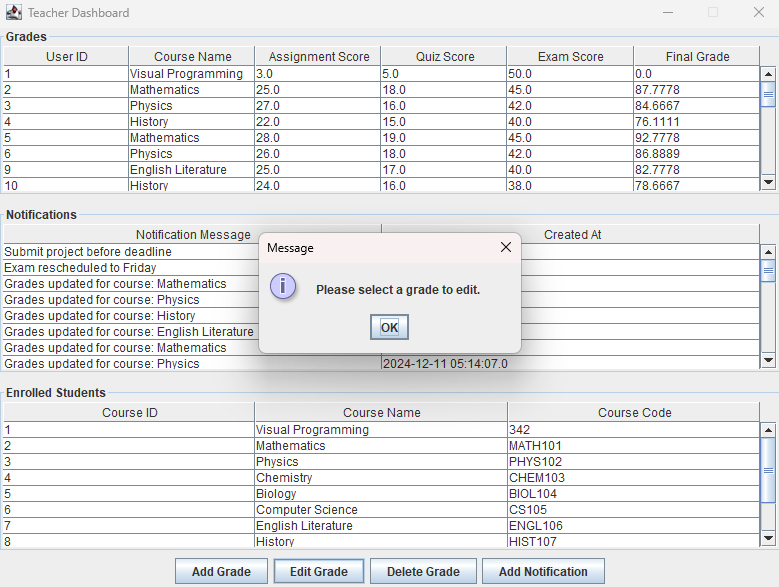
* When there’s an error in the username or password inputs or not valid



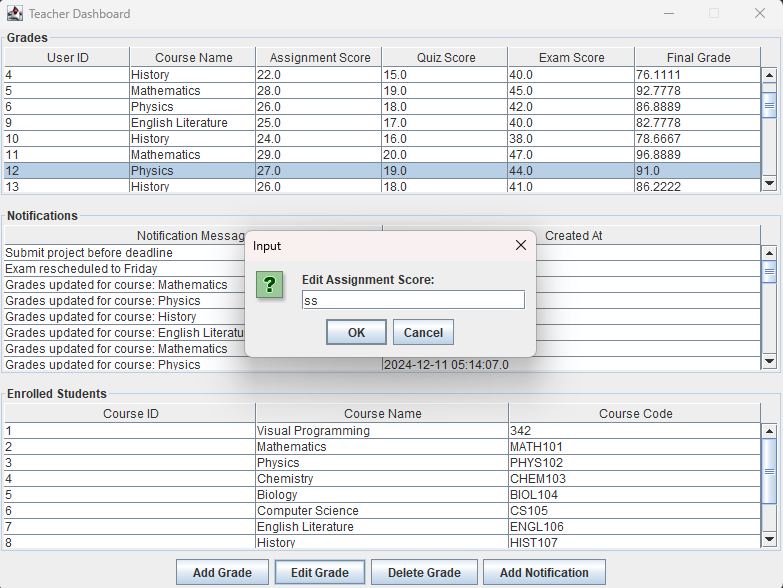
* When there’s an error in the Add Grade inputs (same goes with Edit Grade)

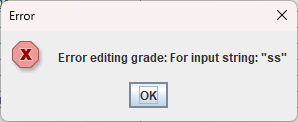


* Not selecting a grade exception (for Edit Grade and Delete Grade)



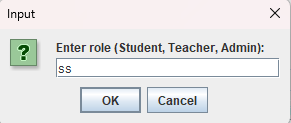
After we select a grade, we try to type unexpected input

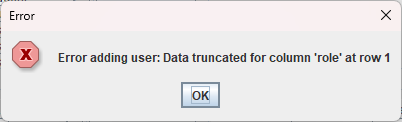




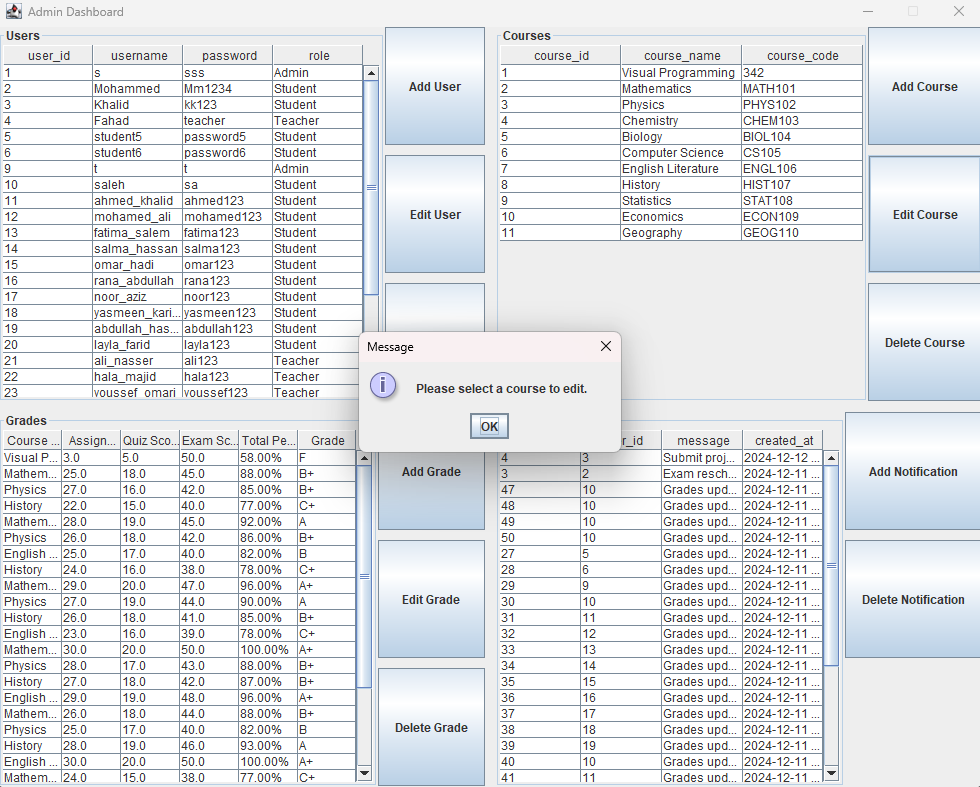
Exceptions for AdminGui are so many, here’s some of them:

* When you input a role other than Student, Teacher, Admin(the letters case doesn’t matter):

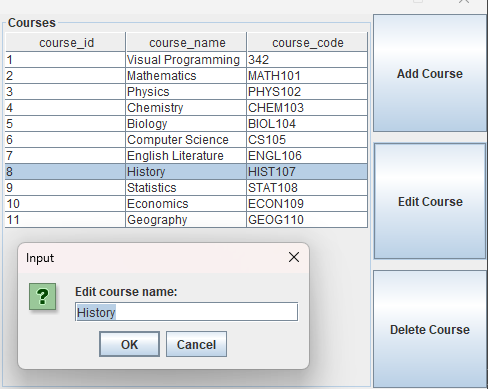


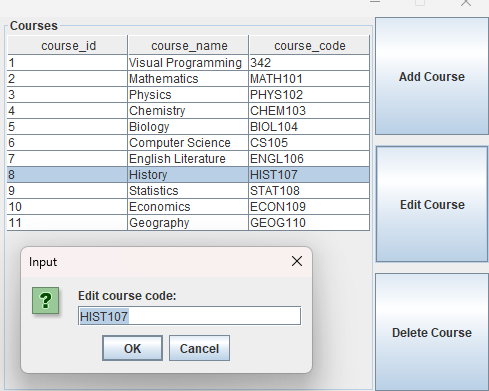


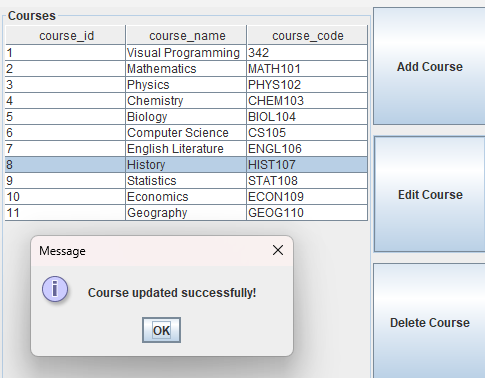
* Edit Grade exceptions (for example):



You can not edit the Course ID, Student ID, or even Grade ID to avoid errors in the database







### 5. Testing and Validation

**1. Functional Testing:**

- Verified authentication, role-based access, grade calculations, and GPA computation.

- Ensured proper handling of inputs, including invalid or missing data.

### 2. Database Validation:

- Tested all CRUD operations (Create, Read, Update, Delete) for users, grades, and notifications.

- Ensured correct data retrieval and error handling for invalid queries.

### 3. GUI Testing:

- Validated all GUI components, including tables, combo boxes, and buttons, to ensure proper display and interactivity.

### 4. Edge Cases:

- Tested missing inputs, invalid row selections, and database errors with clear error messages.

### 4. Performance and Usability:

- **Checked responsiveness with large datasets and ensured user-friendly design and clear instructions.**

**The full Project’s code will be attached on a sperate .txt file**