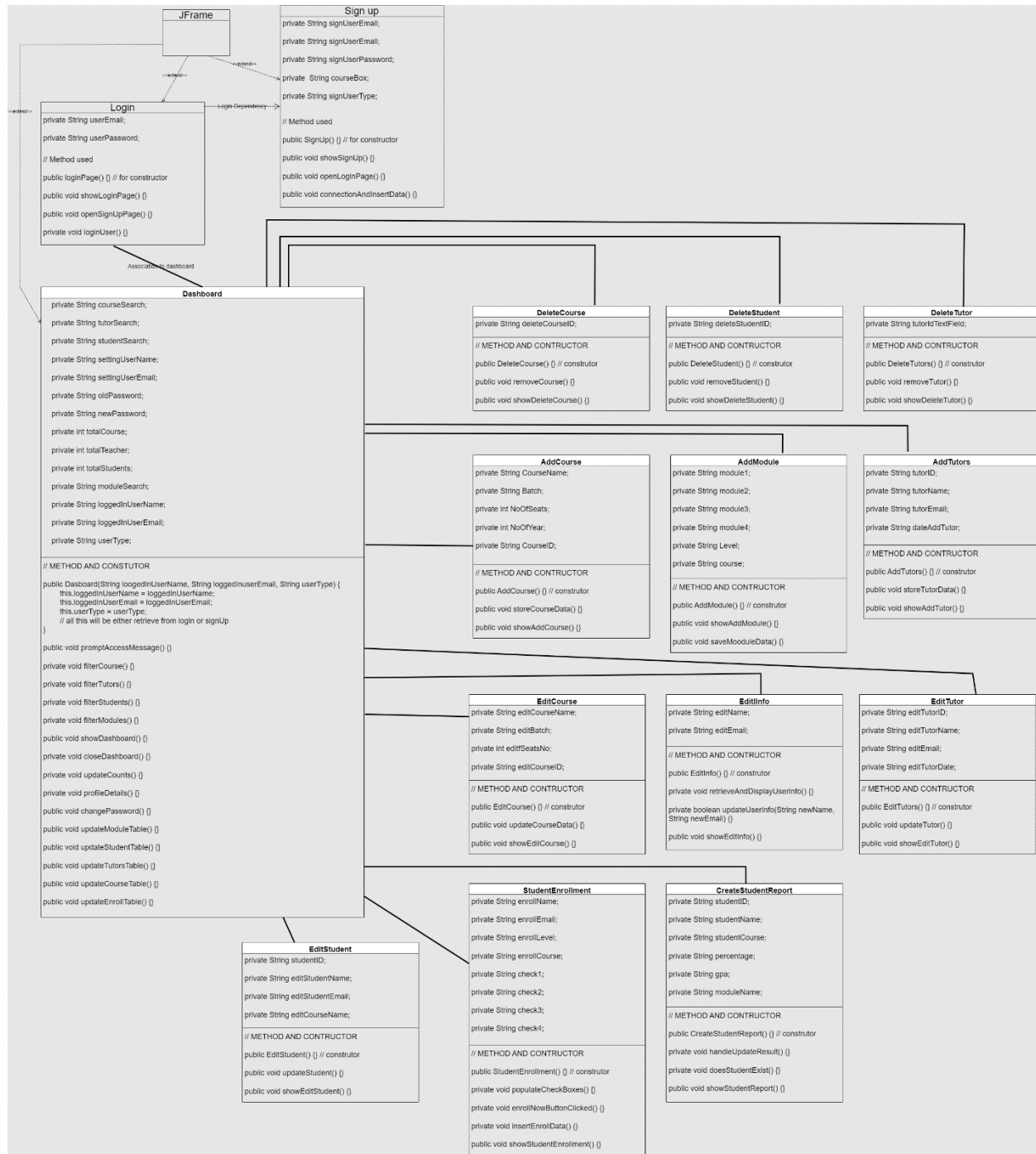


UML Diagram – CMS



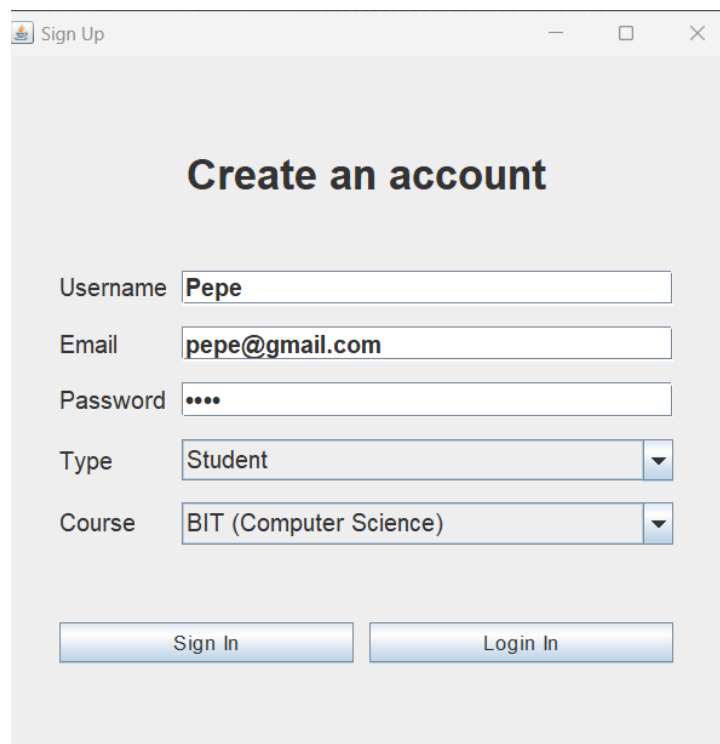
UML – EXPLANATION

UML, or Unified Modeling Language, is like a toolbox for drawing pictures that show how software works. Recently, I used UML to draw a detailed picture of a Course Management System, which helps colleges handle course registrations smoothly.

In this system, students are important because they're the ones who use it the most. The UML diagram I made shows all the different parts of the system and how they fit together. It's like a map that helps everyone understand how the system works and what it does. Below is the class diagram for overall interaction of the system.

SignUp Class Diagram

Class contains field properties like userName, userEmail, userPassword, course, userType etc. All these properties has a datatype of string which is later store in the database. Consists different method to open login / sign up page. Also there is method for connection and insert of data to database with proper validation. If all the field is filled then dashboard panel will open.



Sign Up

Create an account

Username

Email

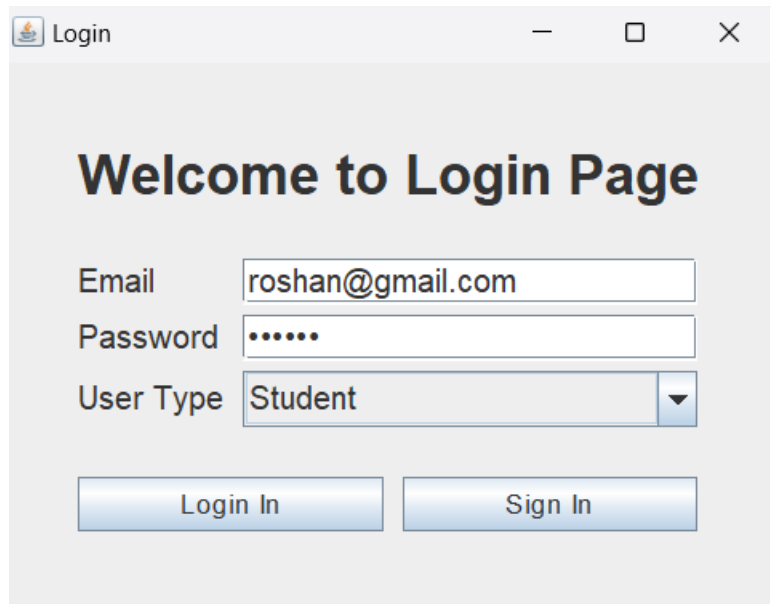
Password

Type

Course

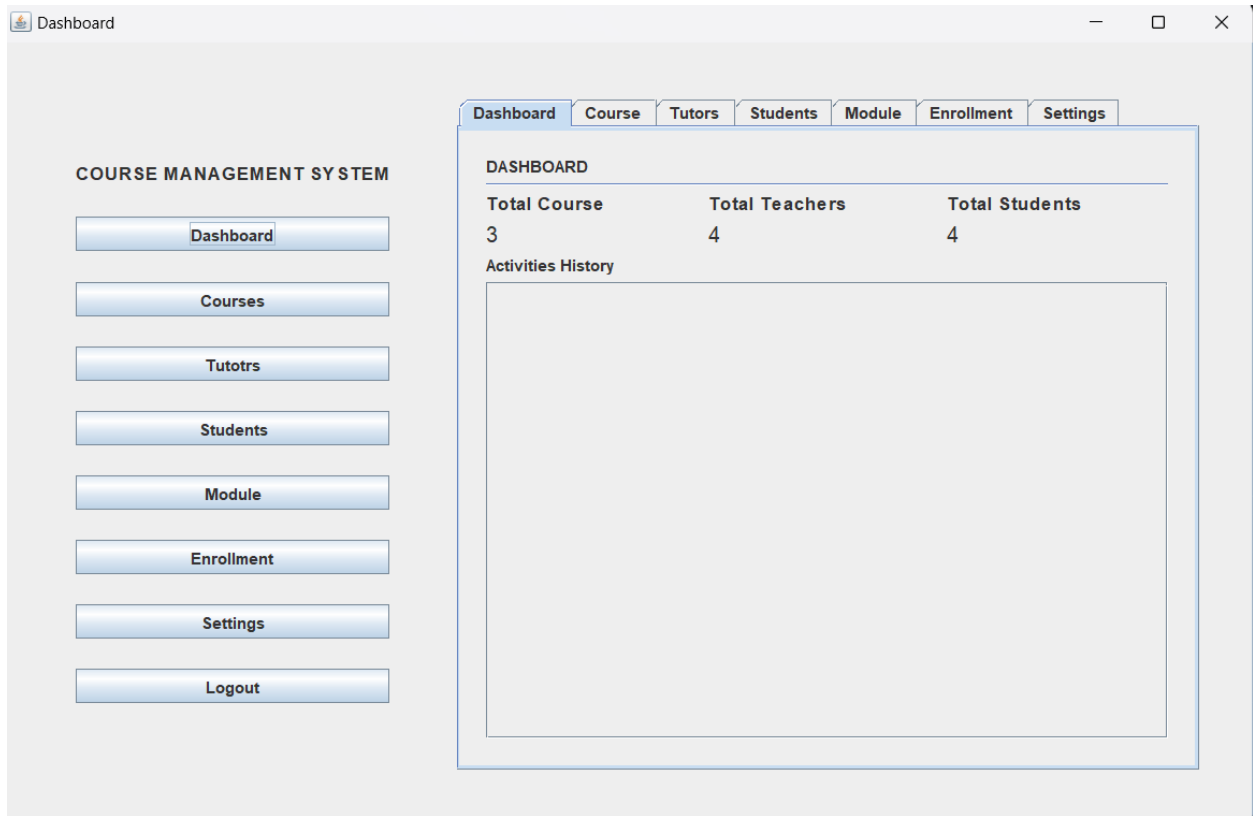
Login Class Diagram

Contains fields like userEmail, userPassword. These fields are input by the user. The input field must match to the data in the database. Login class is also dependent to sing up class. Cause the validation process is only done when the user first signs up. It contains method loginUser which checks all the above discussed validation. And open the dashboard if everything is fine.

A screenshot of a Java Swing window titled "Login". The window has a light gray background and a title bar with standard Windows controls (minimize, maximize, close). The main content area displays "Welcome to Login Page" in a large, bold, black font. Below the title, there are three input fields: "Email" with the text "roshan@gmail.com", "Password" with masked characters "*****", and "User Type" with a dropdown menu showing "Student". At the bottom, there are two buttons: "Login In" and "Sign In", both with a blue gradient and white text.

Dashboard Class Diagram

Main part of the course management system. Dashboard is associated with both login and signup. The data from the login and sign is necessary cause inside the dashboard restriction is built according to the userType. It contains various fields like search, totalCourse, totalTeacher, totalStudents and many more. Various buttons are used to perform certain tasks inside the dashboard. Method to filter all the course, tutor, student, and module according to the search. Method to open the show dashboard UI and close dashboard when logout button is clicked. Method to update table for student, tutor, course, enrollment, module etc.



Add Course Class Diagram

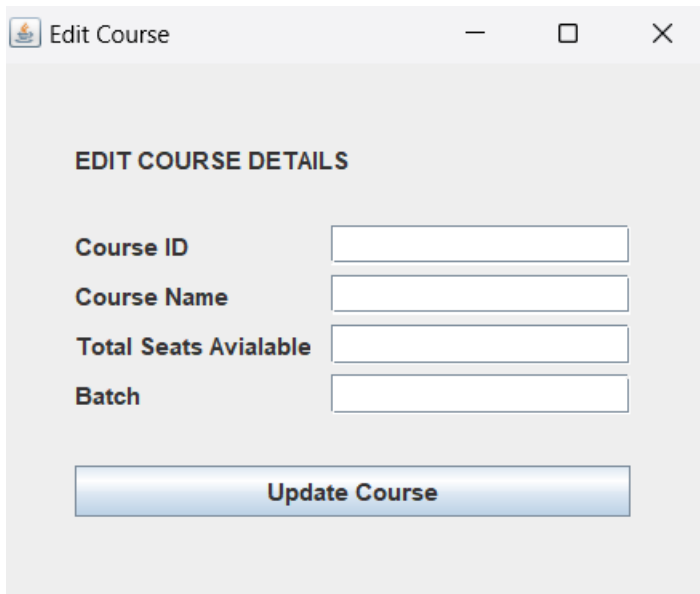
To add the course and update the course database table which can only be done by the admin. It consists of field like courseName, batch, noOfSeats, noOfYear, courseId. Method to store course data.

The screenshot shows a web application window titled "Add Course". It contains a form with the following fields and a button:

- Course ID:
- Course Name:
- Batch:
- Year:
- No. of seats:
-

Edit Course Class Diagram

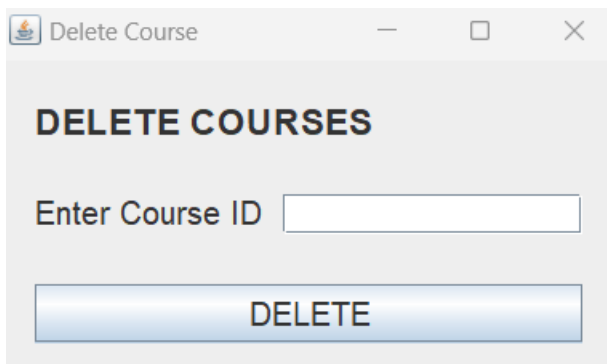
To edit the course and update both in course table and database table. Field like courseID, courseName, totalNoSeats, batch. Method update course data and show course page.



The screenshot shows a window titled "Edit Course" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area has a heading "EDIT COURSE DETAILS". Below this heading, there are four labeled text input fields arranged vertically: "Course ID", "Course Name", "Total Seats Available", and "Batch". At the bottom of the form is a large, light blue button with the text "Update Course".

Delete Course Class Diagram

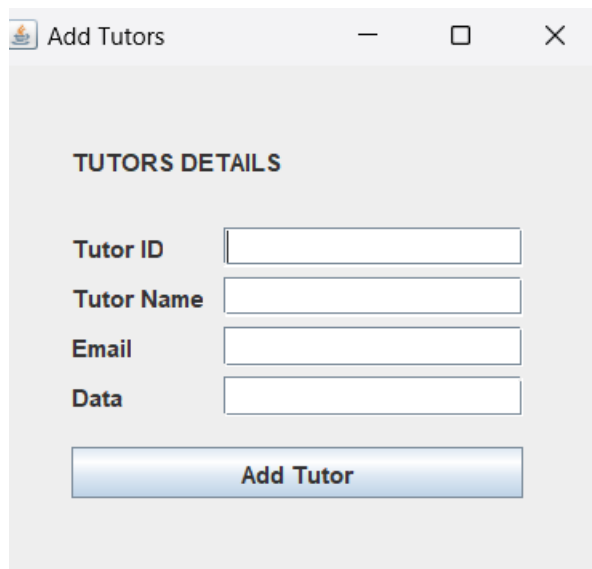
To delete the course permanently from the table. Has field courseID according to that id course store in the database it delete the data. Method removeCourse.



The screenshot shows a window titled "Delete Course" with a standard Windows-style title bar. The main content area has a heading "DELETE COURSES". Below this heading, there is a single labeled text input field with the text "Enter Course ID". At the bottom of the form is a large, light blue button with the text "DELETE".

Add Tutor Class Diagram

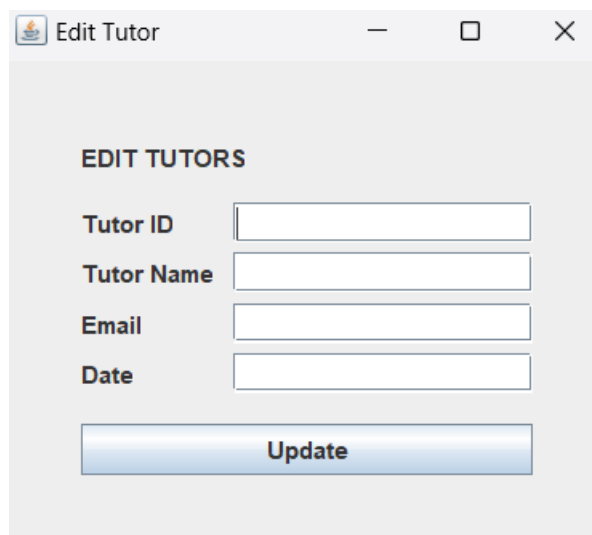
To add tutor teacher and update the teacher table. Consists of field tutorID, tutorName, tutorEmail, dateAddTutor which means the data during the add process of tutor. Method to update the tutor table.



The screenshot shows a window titled "Add Tutors" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a light gray background. At the top, the text "TUTORS DETAILS" is displayed in bold. Below this, there are four labeled text input fields: "Tutor ID", "Tutor Name", "Email", and "Data". Each field is a simple white rectangle with a thin gray border. At the bottom of the form area, there is a blue button with a gradient and the text "Add Tutor" in white.

Edit Tutor Class Diagram

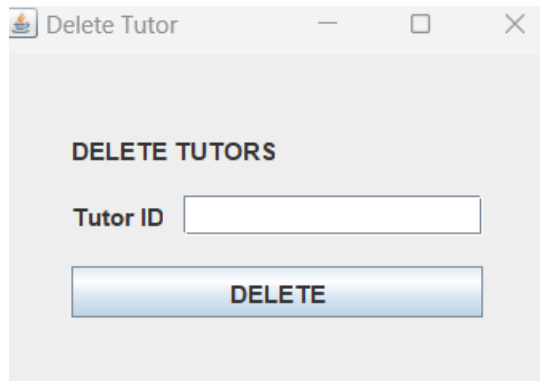
To edit the teacher and update the teacher table. Consists of field editTutorID which won't be edit but check if the enter id exists or not if exit then the remaining name, email, date might be change. Method to update tutor table



The screenshot shows a window titled "Edit Tutor" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a light gray background. At the top, the text "EDIT TUTORS" is displayed in bold. Below this, there are four labeled text input fields: "Tutor ID", "Tutor Name", "Email", and "Date". Each field is a simple white rectangle with a thin gray border. At the bottom of the form area, there is a blue button with a gradient and the text "Update" in white.

Delete Tutor Class Diagram

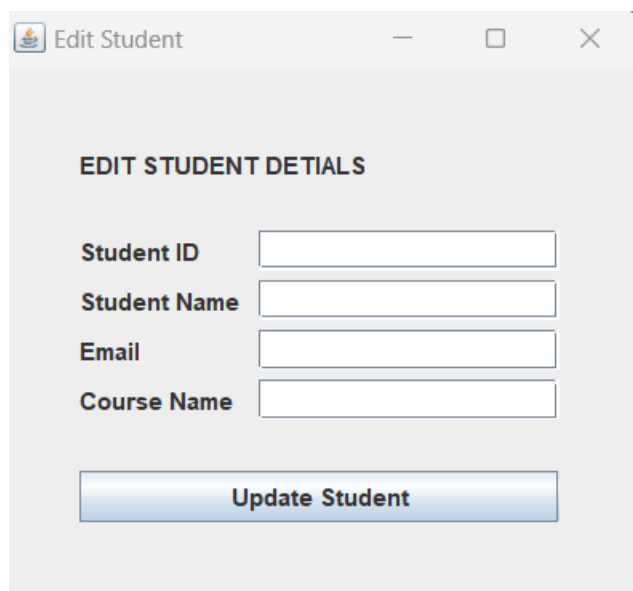
To delete the teacher details permanently from the teacher table. It's has field called deleteTutorID which selects and delete that particular id through method called removeTutor.



A screenshot of a Java Swing window titled "Delete Tutor". The window has a light gray background and standard window controls (minimize, maximize, close) in the title bar. The main content area is titled "DELETE TUTORS" in bold. Below the title, there is a label "Tutor ID" followed by a text input field. At the bottom of the window, there is a blue button with the text "DELETE" in white.

Edit Student

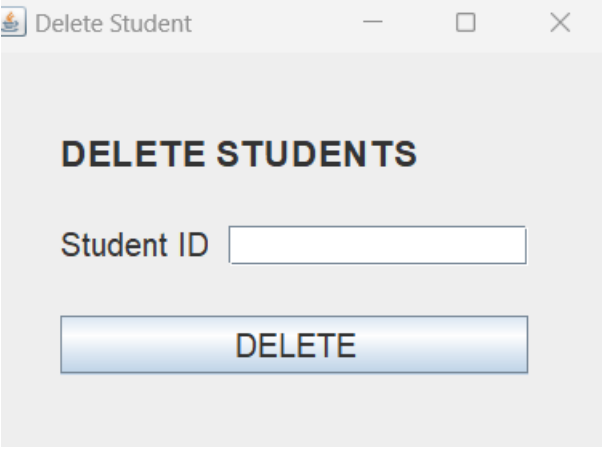
In some scenarios you might want to edit student details. So in this case edit student page is used to edit where students id must match to the database table in order to edit the name, email, course field properties. After this updateStudent method is used to upgrade the student table.



A screenshot of a Java Swing window titled "Edit Student". The window has a light gray background and standard window controls (minimize, maximize, close) in the title bar. The main content area is titled "EDIT STUDENT DETIALS" in bold. Below the title, there are four labels with corresponding text input fields: "Student ID", "Student Name", "Email", and "Course Name". At the bottom of the window, there is a blue button with the text "Update Student" in white.

Delete Student

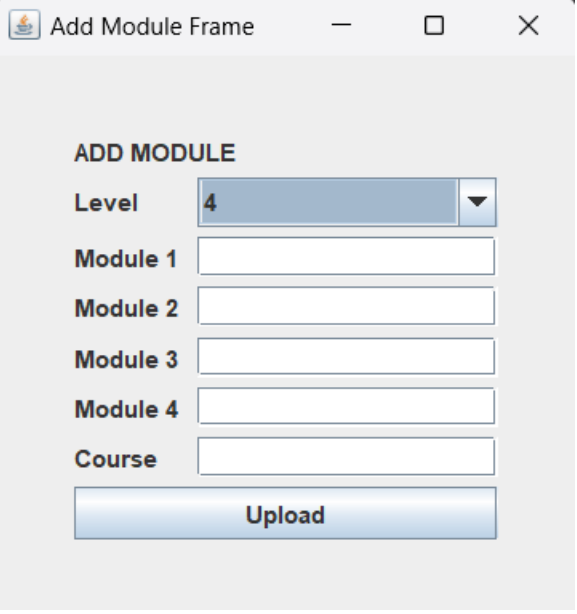
To delete the student permanently from the table. There is filed properties deleteStudentID. The id is checked and if it exists in the table then it is removed from the table through method removeStudent.



A screenshot of a Java Swing window titled "Delete Student". The window has a light gray background and a title bar with standard Windows controls. The main content area features the text "DELETE STUDENTS" in bold, black, uppercase letters. Below this, there is a label "Student ID" followed by a text input field. At the bottom of the window is a large, blue, rectangular button with the word "DELETE" in white, uppercase letters.

Add Module

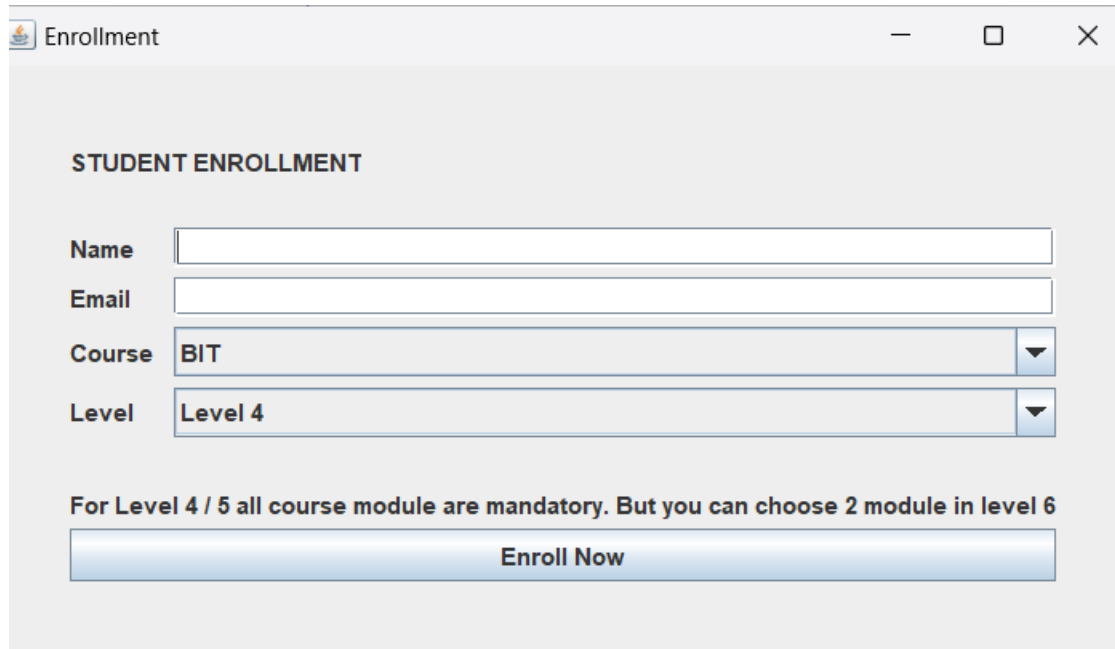
To add the module. It has field level, module1, module2, module3, module4, course and all the data is sent to the database table through method called saveModuleData.



A screenshot of a Java Swing window titled "Add Module Frame". The window has a light gray background and a title bar with standard Windows controls. The main content area features the text "ADD MODULE" in bold, black, uppercase letters. Below this, there are five labels: "Level", "Module 1", "Module 2", "Module 3", and "Module 4", each followed by a text input field. The "Level" field is a dropdown menu with the number "4" selected. Below these fields is a label "Course" followed by a text input field. At the bottom of the window is a large, blue, rectangular button with the word "Upload" in white, uppercase letters.

Enroll

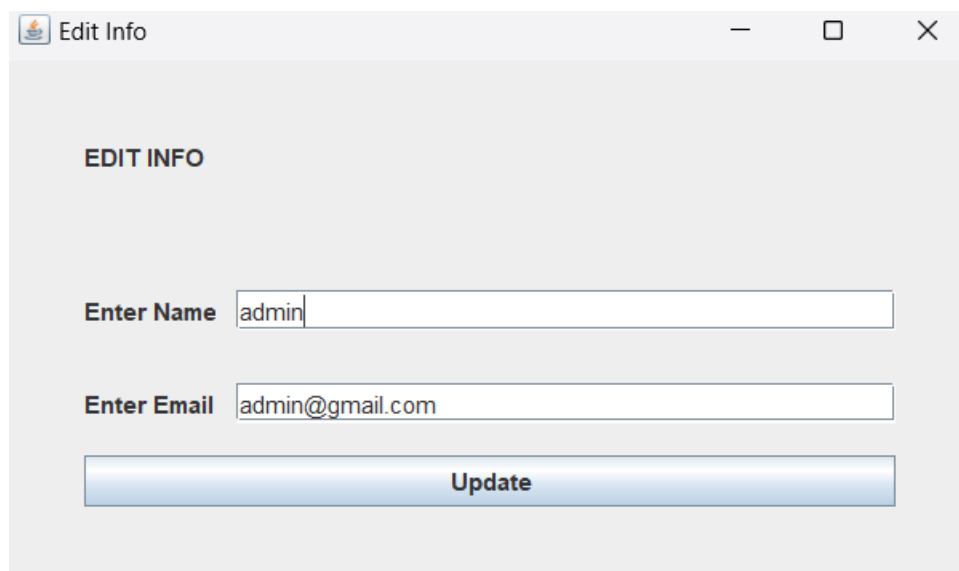
To enroll the student and it require field like name, email, course, level. All these is done through method StudentEnrollment.



The screenshot shows a web application window titled "Enrollment". Inside the window, there is a section titled "STUDENT ENROLLMENT". Below this title, there are four input fields: "Name" (a text box), "Email" (a text box), "Course" (a dropdown menu with "BIT" selected), and "Level" (a dropdown menu with "Level 4" selected). Below these fields, there is a line of text: "For Level 4 / 5 all course module are mandatory. But you can choose 2 module in level 6". At the bottom of the form is a large blue button labeled "Enroll Now".

Edit Info

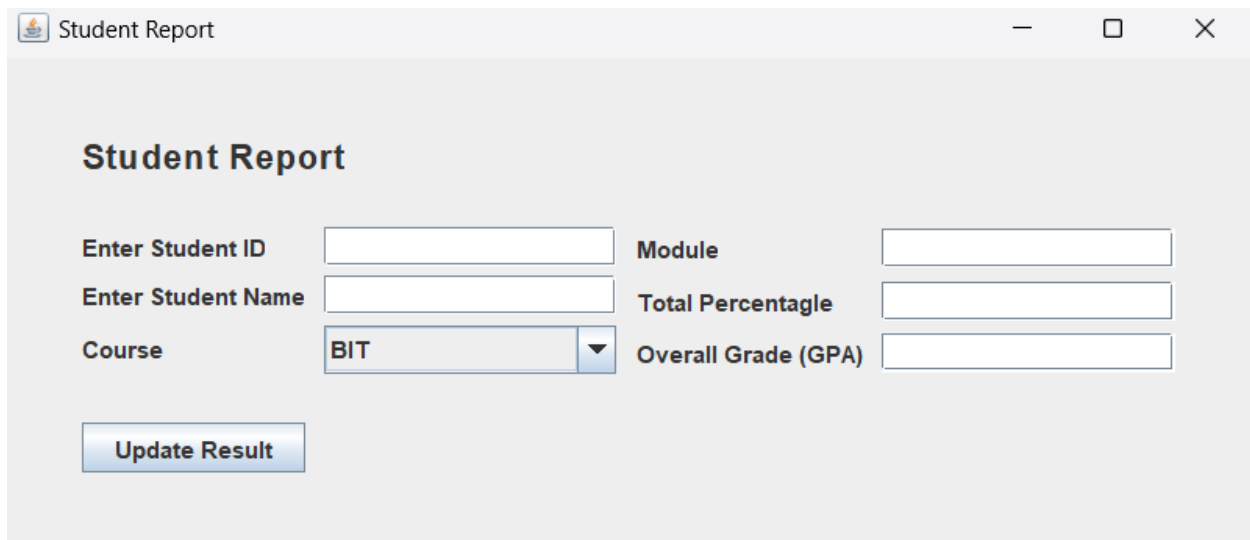
To edit the info of the logged in user. Has field name of editUserName, editUserEmail which will later update the data in the table also.



The screenshot shows a web application window titled "Edit Info". Inside the window, there is a section titled "EDIT INFO". Below this title, there are two input fields: "Enter Name" (a text box containing "admin") and "Enter Email" (a text box containing "admin@gmail.com"). At the bottom of the form is a large blue button labeled "Update".

Create Student Report

To create student report according to the module performance and this is done by the tutor. Field studentID, studentName, course, module, totalPercentage, gpa. Using all these field value a method handleUpdateResult is used to create student report.



Student Report

Enter Student ID Module

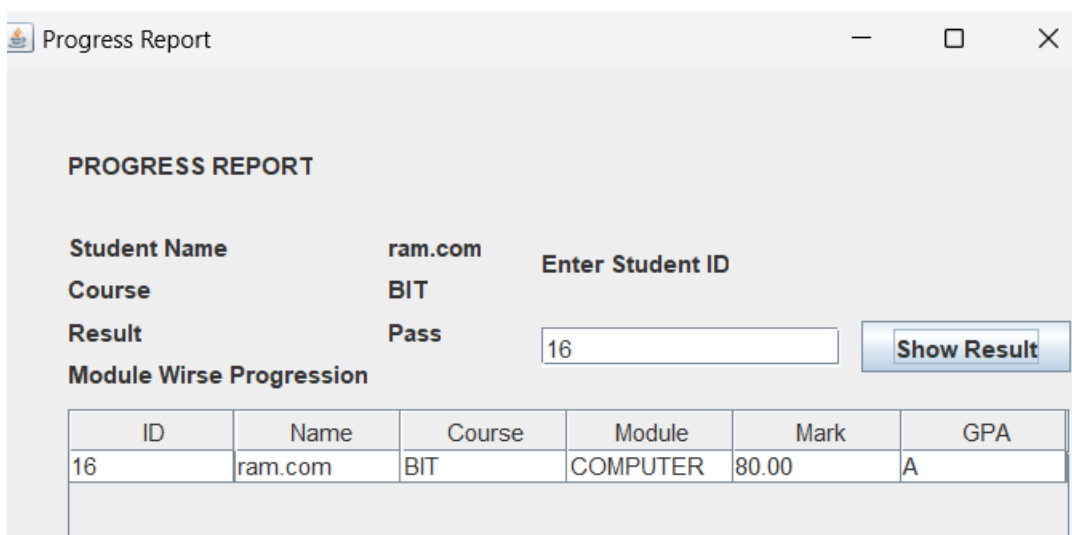
Enter Student Name Total Percentage

Course Overall Grade (GPA)

Update Result

View Project

To view the result progress of each student report in their respective module. The report created by the teacher will view the view progress table and access the data from the database table.



PROGRESS REPORT

Student Name ram.com Enter Student ID
Course BIT
Result Pass
Module Wise Progression 16 **Show Result**

ID	Name	Course	Module	Mark	GPA
16	ram.com	BIT	COMPUTER	80.00	A

Conclusion

To conclusion, using UML to draw the Course Management System diagram has made it easy to understand how everything works together. It shows how students interact with the system and helps everyone involved see the big picture. This diagram is like a roadmap for building and using the system, making course registrations at colleges smoother and more organized.