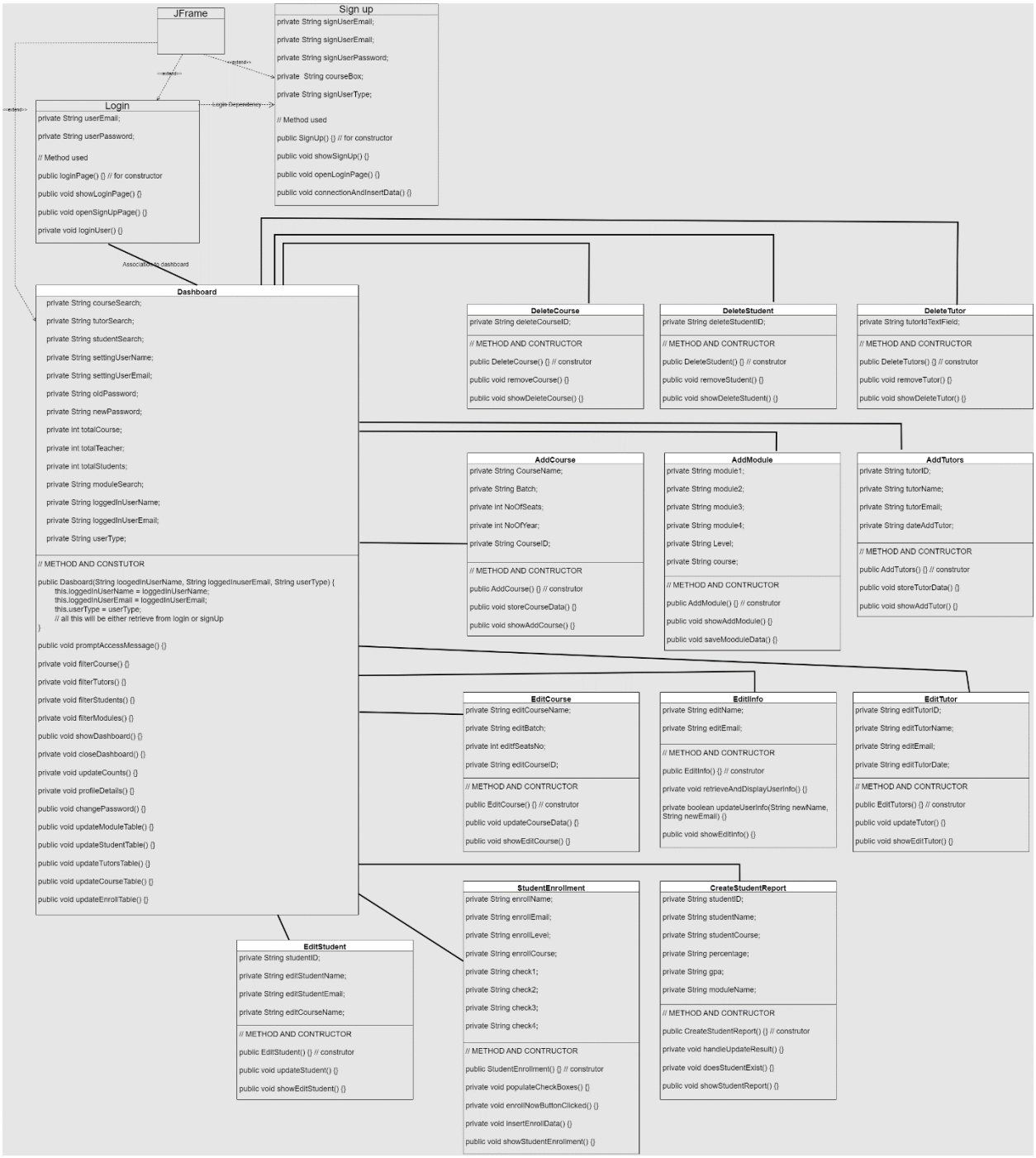
UML Diagram – CMS



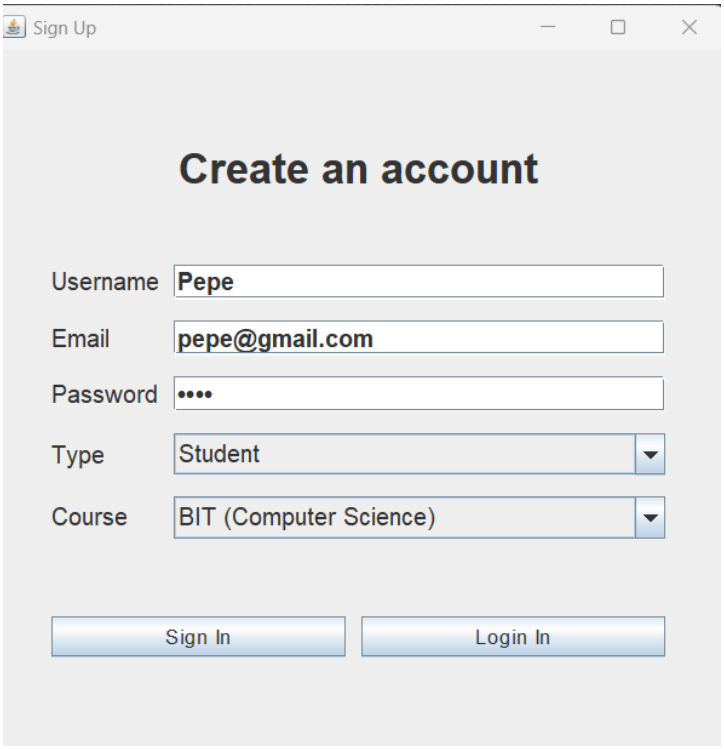
UML – EXPLAINATION

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.

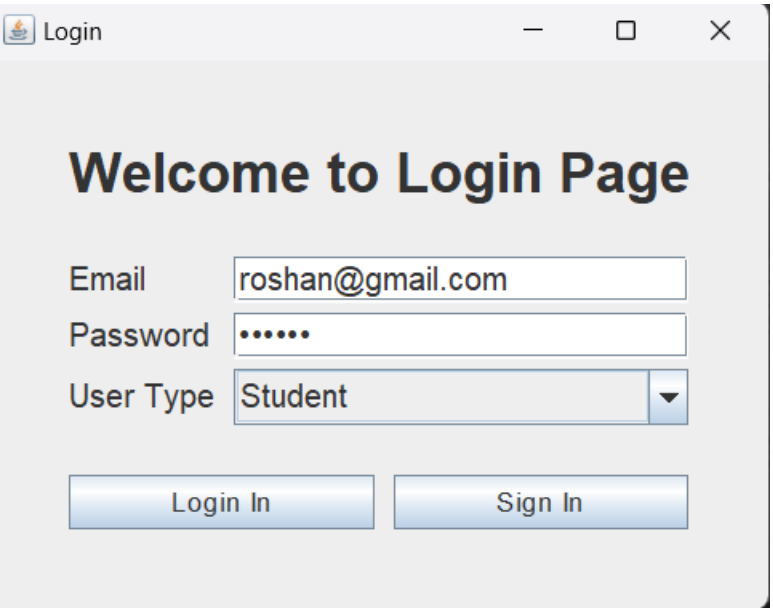
SignUp Class Diagram

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.



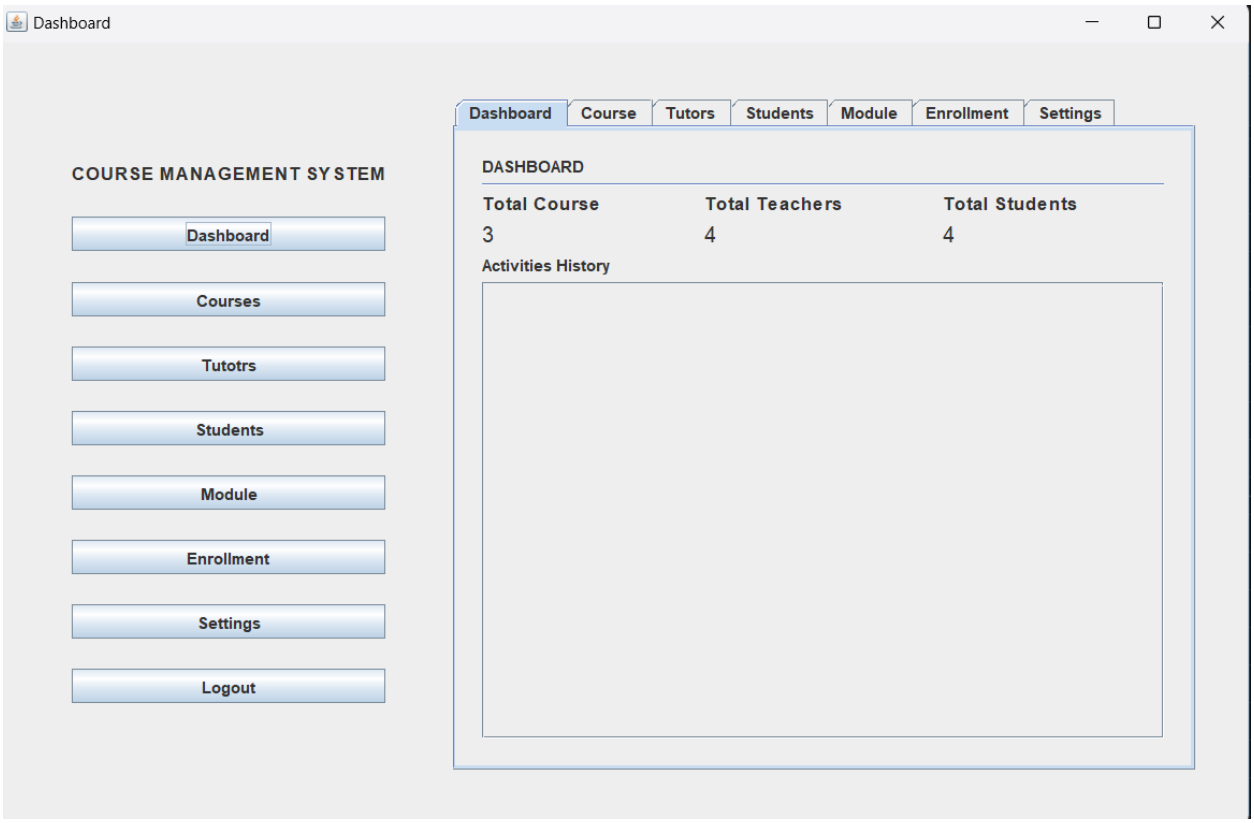
Login Class Diagram

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.



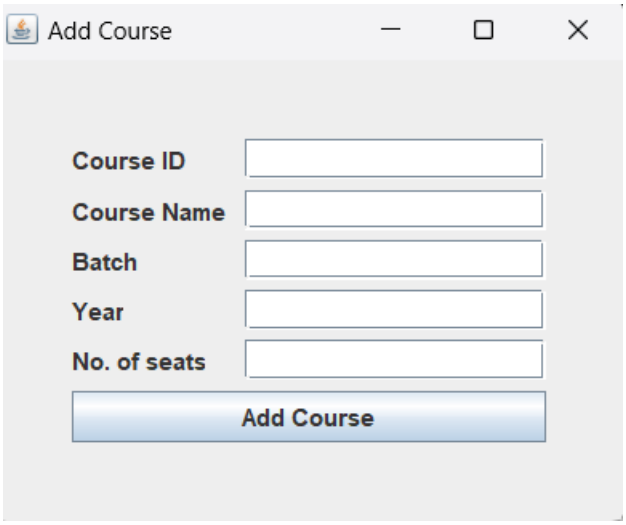
Dashboard Class Diagram

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.



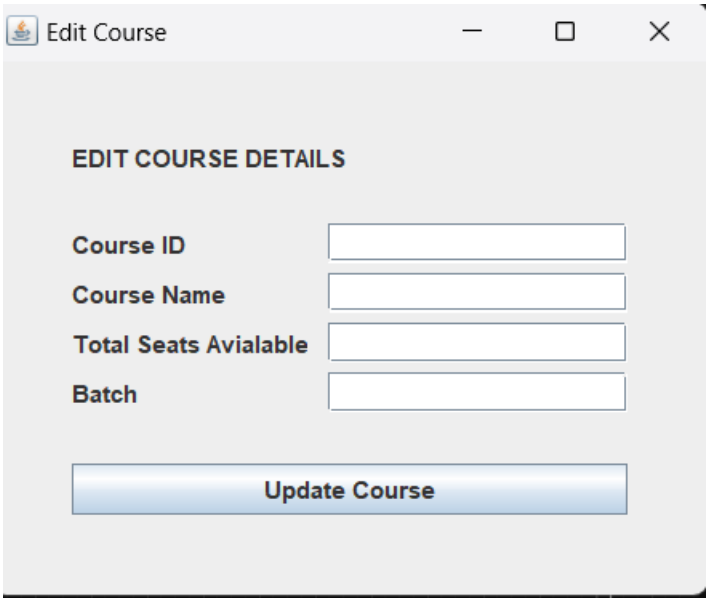
Add Course Class Diagram

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.



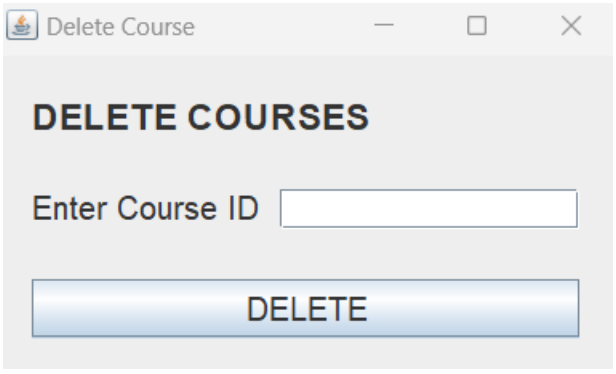
Edit Course Class Diagram

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.



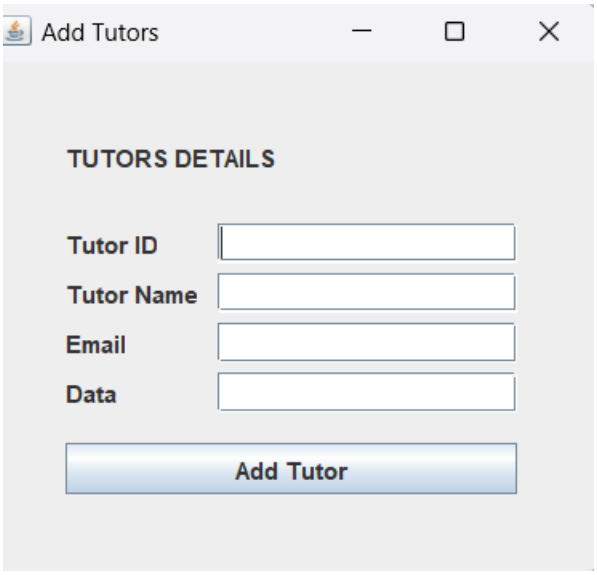
Delete Course Class Diagram

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.



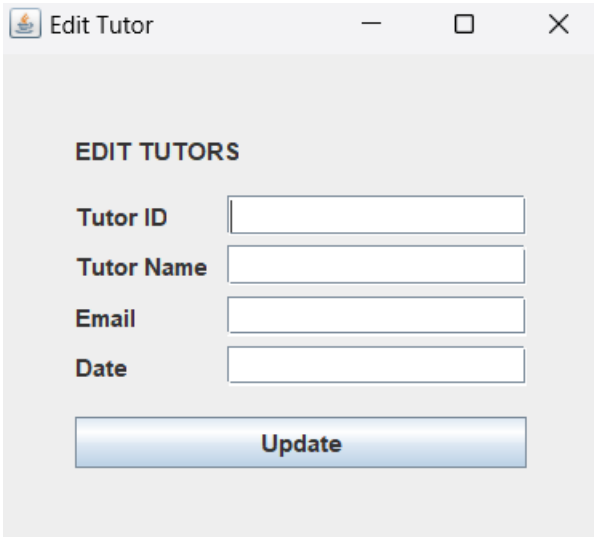
Add Tutor Class Diagram

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.



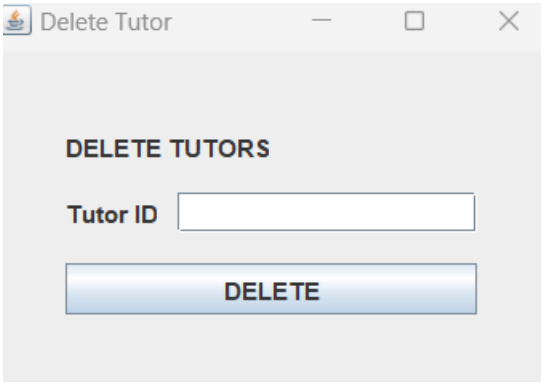
Edit Tutor Class Diagram

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



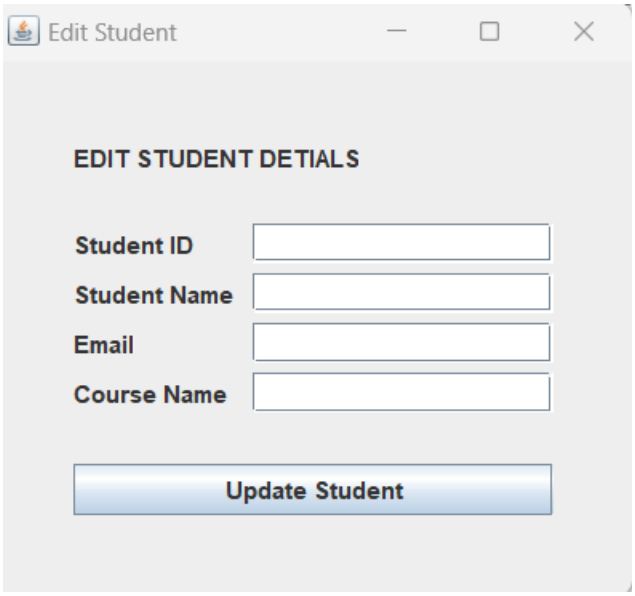
Delete Tutor Class Diagram

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.



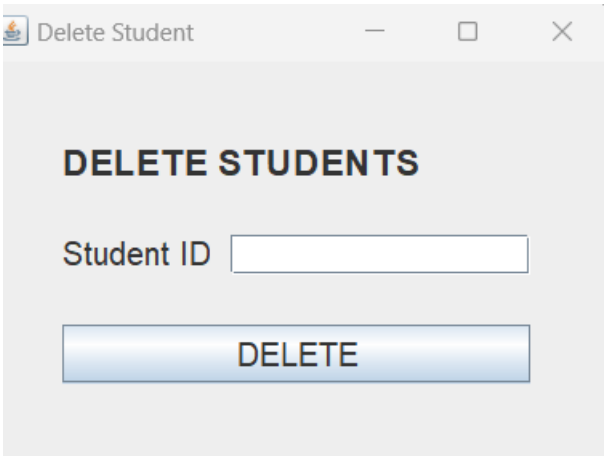
Edit Student

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.



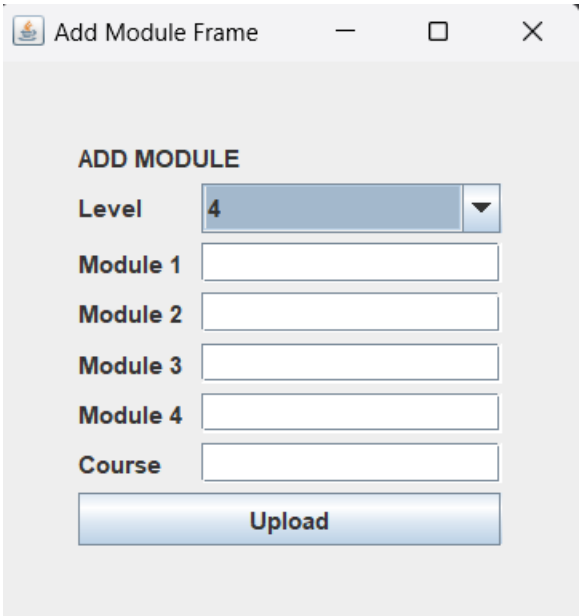
Delete Student

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.



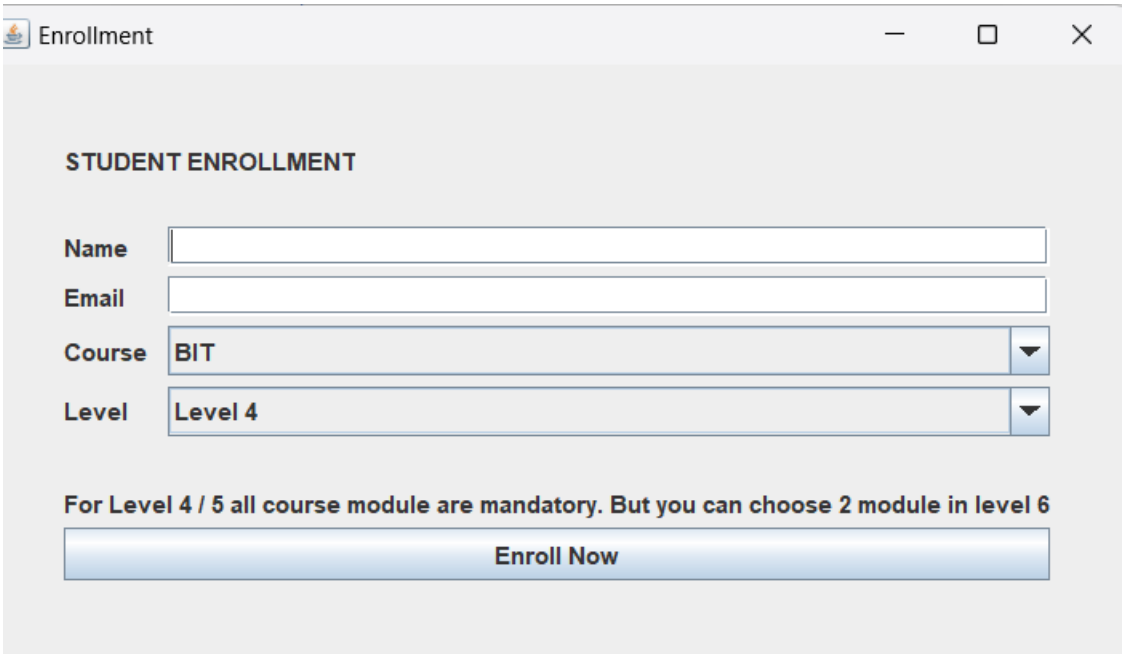
Add Module

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.



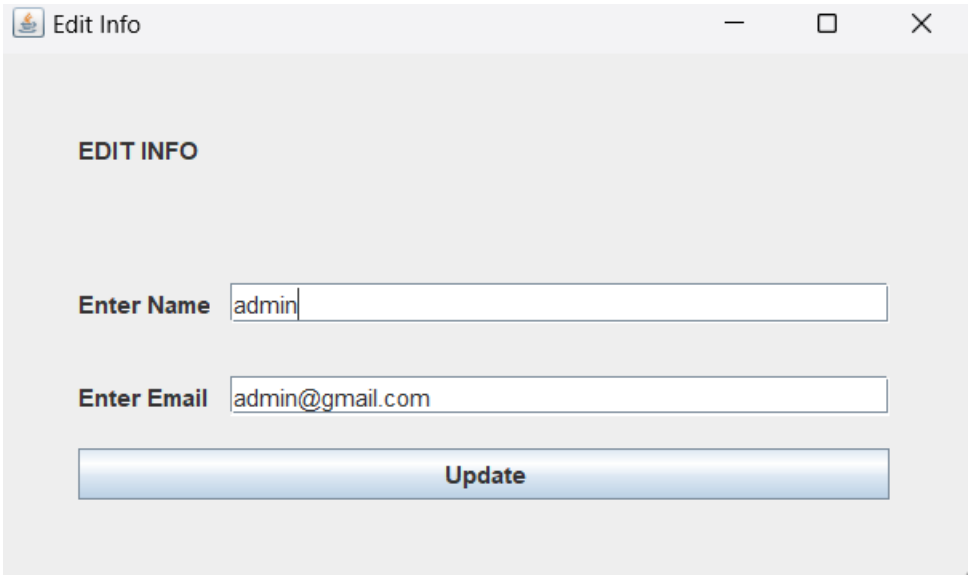
Enroll

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



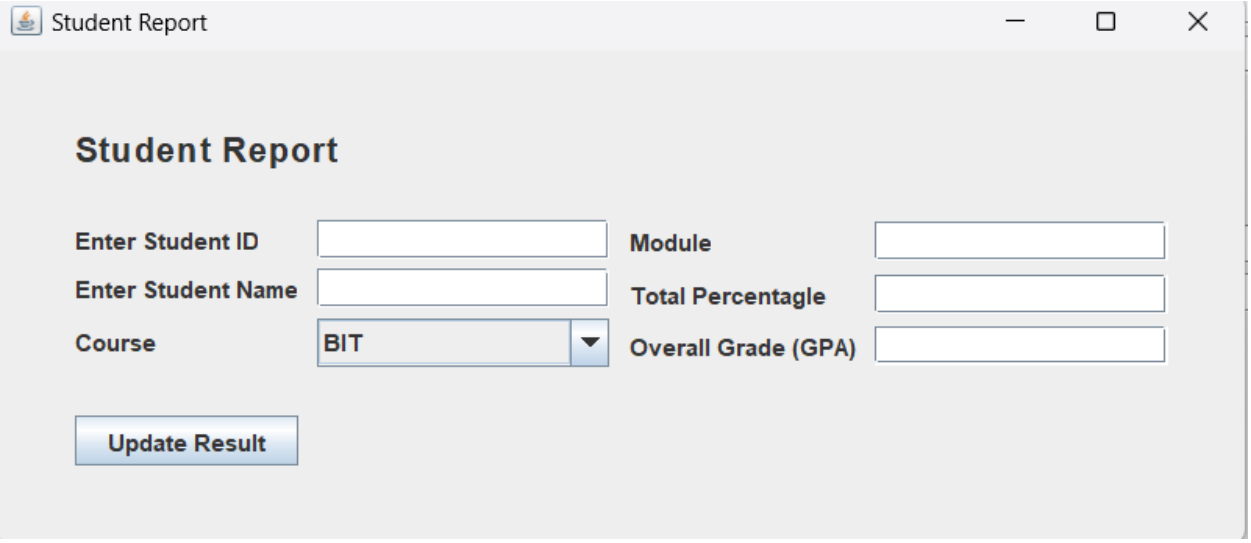
Edit Info

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



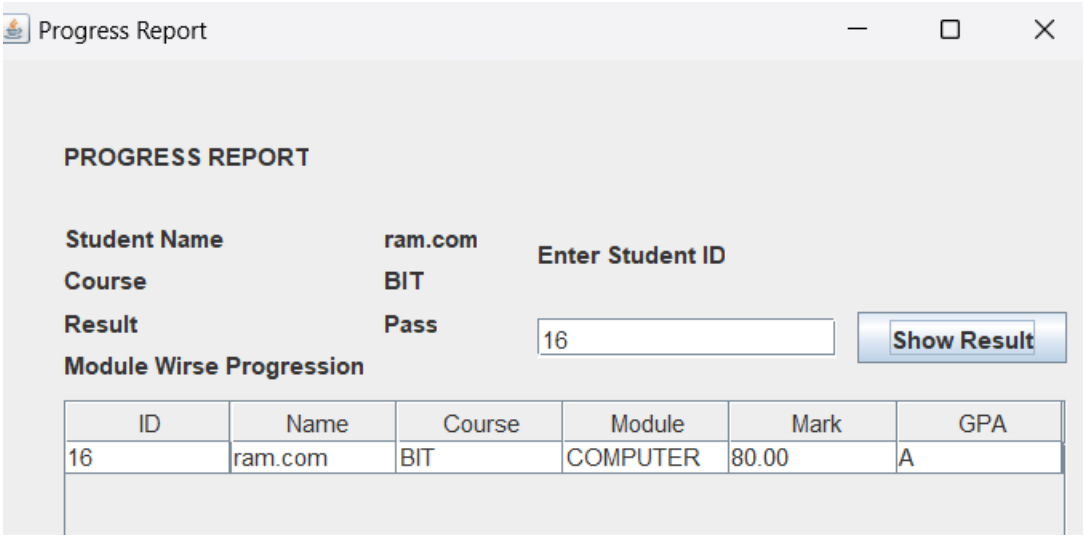
Create Student Report

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.



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.



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.