UML Class Diagram

Login
<<Java Class>>
Login
coursemanagement

- frame: JFrame
- login_student: JPanel
- home: JPanel
- txtEmail: JTextField
- lblNew Label: JLabel
- passwordField: JPasswordField
- Signup: JPanel
- txtName: JTextField
- signup_password: JPasswordField
- teacher_btn: JButton
- login_teacher: JPanel
- teacher_email: JTextField
- teacher_pass: JPasswordField
- teacher_code: JPasswordField
- adminEmail: JTextField
- adminPassword: JPasswordField
- adminCode: JPasswordField
- adminbtn: JButton
- login_Admin: JPanel
- signup_mail: JTextField
- levelField: JTextField

- main(String[]):void
- Login(DatabaseManageR)
- initialize():void
- setVisible(boolean):void

Dashboard
<<Java Class>>
Dashboard
coursemanagement

- role: String
- DashboardFrame: JFrame
- table: JTable
- teachers_table: JTable
- course_table: JTable
- module_table: JTable

- main(String[]):void
- Dashboard(DatabaseManageR,String)
- initialize():void

EditTeachers
<<Java Class>>
EditTeachers
coursemanagement

- frame: JFrame
- teachername: JTextField
- Teacher_email: JTextField
- add: JLabel
- id_ic: JLabel
- name_ic: JLabel
- trash: JLabel
- layeredPane: JLayeredPane
- code: JPasswordField
- phoneno: JTextField
- password: JPasswordField
- usererror: JLabel
- emailerror: JLabel
- pherror: JLabel
- passerror: JLabel
- password1: JTextField
- lblNew Label: JLabel

- main(String[]):void
- EditTeachers()
- initialize(JLayeredPane):void

GetResult
<<Java Class>>
GetResult
coursemanagement

- frame: JFrame
- table: JTable

- main(String[]):void
- GetResult(DatabaseManageR)
- initialize():void

GenerateResult
<<Java Class>>
GenerateResult
coursemanagement

- frame: JFrame
- student_id: JTextField
- lblNew Label_1: JLabel
- module_id: JTextField
- marks: JTextField
- lblNew Label: JLabel
- add_report: JButton

- main(String[]):void
- GenerateResult()
- initialize():void

EditCourses
<<Java Class>>
EditCourses
coursemanagement

- frame: JFrame
- course_name: JTextField

- main(String[]):void
- EditCourses()
- initialize():void

Regex
<<Java Class>>
Regex
coursemanagement

- Regex()
- validateEmail(String):boolean
- validatePassword(String):boolean
- validateUsername(String):boolean
- validatePhoneNum(String):boolean
- validateLevel(String):boolean

DatabaseManageR
<<Java Class>>
DatabaseManageR
coursemanagement

- conn: Connection
- stmt: Statement

- getInstance():DatabaseManageR
- DatabaseManageR(String,String,String)
- createDatabase(String):void
- createAuthTable():void
- getConnection():Connection
- getPreparedStatement(String):PreparedStatement

Authenticate
<<Java Class>>
Authenticate
coursemanagement

- retrieveTeachernameStmt: PreparedStatement
- retrieveStudentnameStmt: PreparedStatement
- retrieveadminnameStmt: PreparedStatement
- checkStudentEmailExistenceStmt: PreparedStatement
- checkTeacherEmailExistenceStmt: PreparedStatement
- checkAdminEmailExistenceStmt: PreparedStatement
- addStudentCredentialStmt: PreparedStatement
- addTeacherCredentialStmt: PreparedStatement
- addAdminCredentialStmt: PreparedStatement
- retrieveStudentdetailStmt: PreparedStatement
- addCourseStmt: PreparedStatement
- deleteCourseStmt: PreparedStatement
- addTeacherStmt: PreparedStatement
- deleteTeacherStmt: PreparedStatement
- addReportStmt: PreparedStatement

- Authenticate(DatabaseManageR)
- login(String,String,String,String):void
- addCredential(String,String,String,String,String,String,String,String):void
- updateStudentTable(String,String,String,String,JTable):void
- addCourse(String):void
- deleteCourse(String):void
- addTeacher(String,String,String,String,String,String):void
- deleteTeach(String):void
- addReport(String,String,String):void

Relationship labels:
- ~layeredPane 0..1
- ~db 0..1
- -instance 0..1
- ~db 0..1
- ~db 0..1

Course Management System:

Course Management software program was created to help the faculty, students, and administration. Teachers, students, and administrators can communicate and share information on the course and module content, grades, and other topics on this platform. This software is built using the help of Java Swing in Ecplise IDE and using XAMPP for the database. A total of 9java files and 1 database consisting of 6 tables were implemented for the completion of the task.

Access Modifiers:

private: Accessible only within the class.

default: Accessible only within the package.

protected: Accessible within the package and outside the package but through inheritance only.

public: Accessible everywhere

1.Login.java:

It involves the login and sign-up for the course management system with 3 roles included to login as Student, teacher, admin. Teachers and Admins require special generated code to signup or login it is a bit redundant for the concept as I was trying to involve Verification system. Access modifiers are 21 private fields and It has a public main method along with one private void function.

It has a main class Form. It depends on Dashboard, Regex, Auth and DatabaseManageR.java.

2.Dashboard.java:

It is the main body of the application. Mostly all activities are carried out from here. Access modifiers are 4private fields along with 2 default fields. It has a main class Dashboard() which takes the instance, name and role. It depends on Login, GetResult, EditCourses and GenerateResult.java.

3.DatabaseManageR.java:

It is the back body of the application. It carries out the creation and insertion of databases and tables. Access modifiers are 2 and 4 public methods and 1 private void methods and 1 main class named DatabaseManageR.It has a main class DatabaseManageR() which takes the path, username and password of the XAMPP server.

4.Authenticate.java:

It is the body which contains all the statements and methods that are carried out in the application. Access modifiers are 15,8 public static methods having one main class Authenticate() which takes db from the DatabaseManageR.java. It has 7 public static methods and 3 public methods. It depends on DatabaseManageR.java and Dashboard.java

5.Generate Result:

It is where the teacher can view and generate the report. Access modifiers are 7private variables . It contains a main class GenerareResult() which takes db from the DatabaseManageR.java. It depends on Authenticate, DatabaseManageR.java.

6.GetResult.java:

It is where the student can view the report. Access modifiers are 2 private variables and have one main class GetResult() which takes db from the DatabaseManageR.java and 1 constructor.

7.Regex.java:

It is the main file for validating or filtering unwanted entries. It has a main class Regex() and contains 3 public static methods.

8.EditCourses:

Here the user who has admin privileges can edit the course. Access modifiers are 2private variables along with one main class EditCourses(). It depends on Authenticate.java.

9.EditTeacher

Here the user who has admin privileges can edit the teacher. Access modifiers are 17 private variables are 14 along with one main class EditTeachers(). It depends on Authenticate.java and Dashboard.java.