

# CS5200 Project Final Report

Group name: LiJLiLZhaoZ

## 1. README

- (1) Open the directory in an IDE (make sure you download IDEA and have a java environment)
- (2) Make sure mysql-connector-java.jar is added to dependency
- (3) Run the finalDatabase\_schema.sql in MySQL workbench and make sure the database is added to your scheme
- (4) Find "CourseMain.java" (in window directory), change the account, password and dbName parameters based on your database setup, otherwise the program cannot be connected to the database.
- (5) Run "CourseMain.java" and a log in window will be started. You can use the accounts in Instruction.txt or create new accounts in administration mode.

## 2. Technical specifications

For every education or course there should be consistent interaction between students for registering courses. In this exiting system students only can register a course that instructor wants to teach in current semester after the administrator approves this course.

The system has been developed considering single quality factor. Due to this reason the system is reliable and attributable to the safety features, provides for the course management system, unauthorized users cannot access the system.

### Project Module Summary:

Administrator Module: Admin can produce accounts for college students and faculties and make course programmed list.

Students Module: Student can register with application or the proposed system and login with username and password.

Instructor Module: It can assign grades for courses and submit course applications.

### Software Requirements:

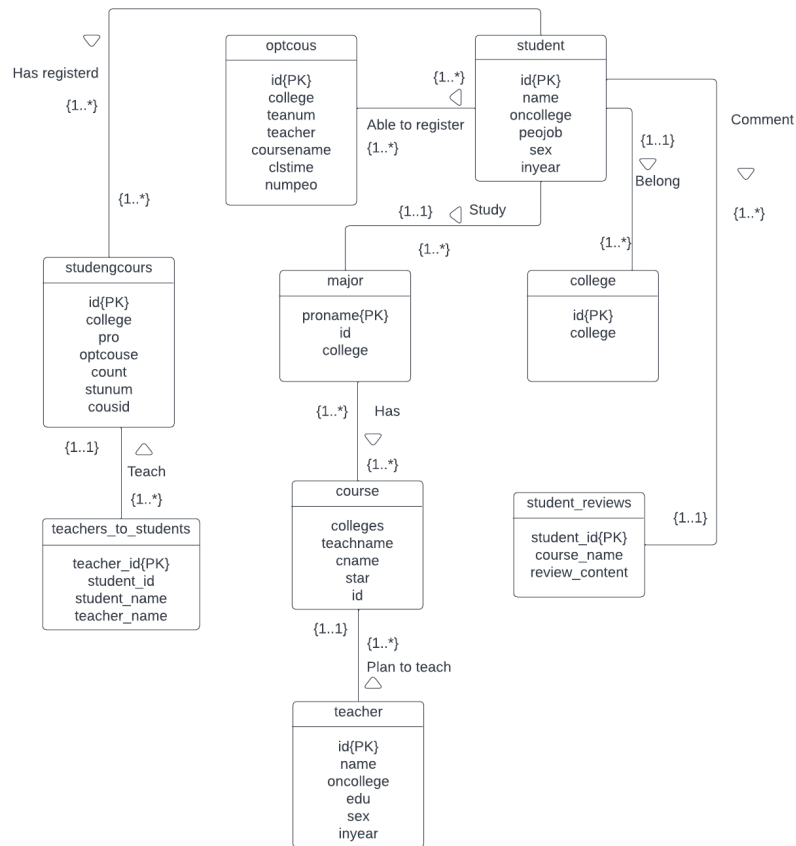
Operating System: MACOS.

Programming Language: Java.

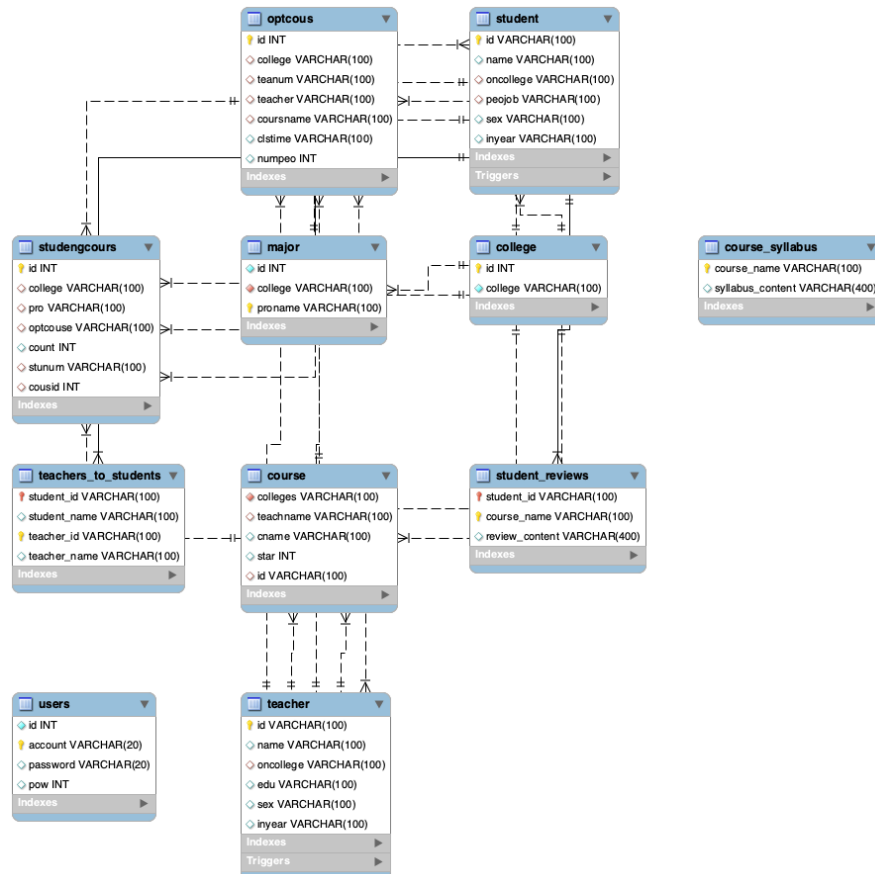
IDE Used: [IntelliJ IDEA](#)

Database: MySQL.

## 3. Conceptual design

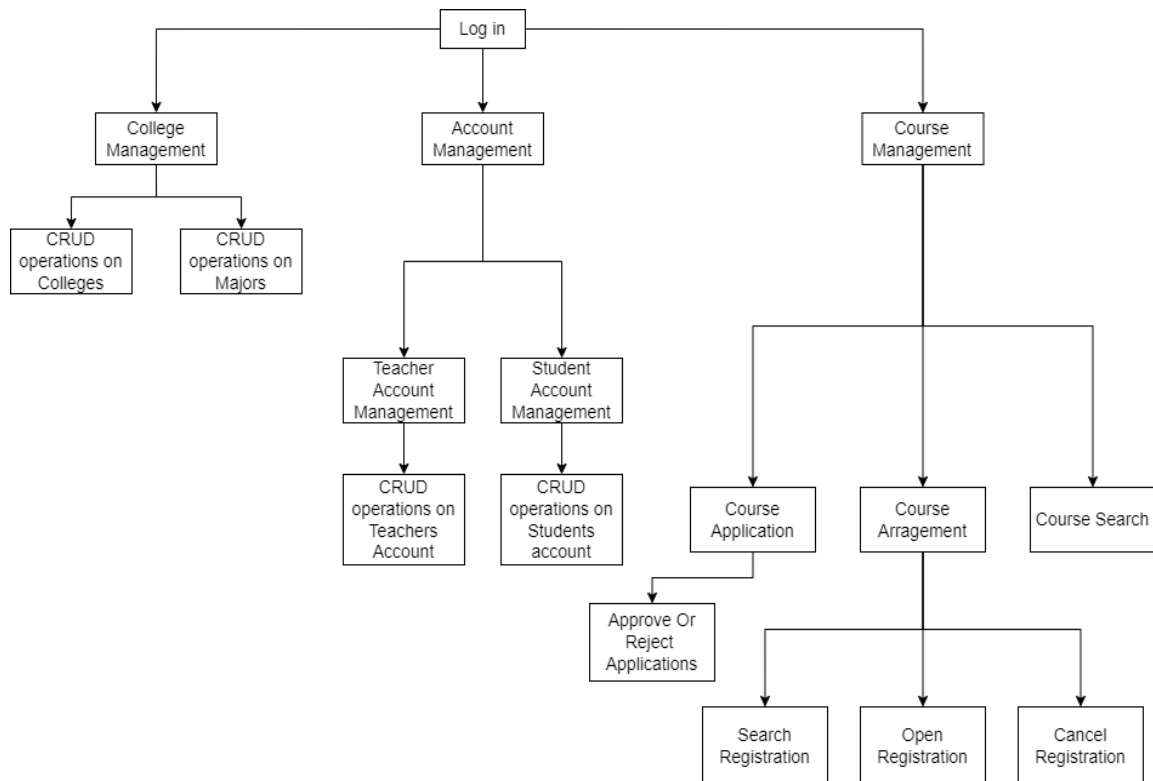


## 4. Logical design

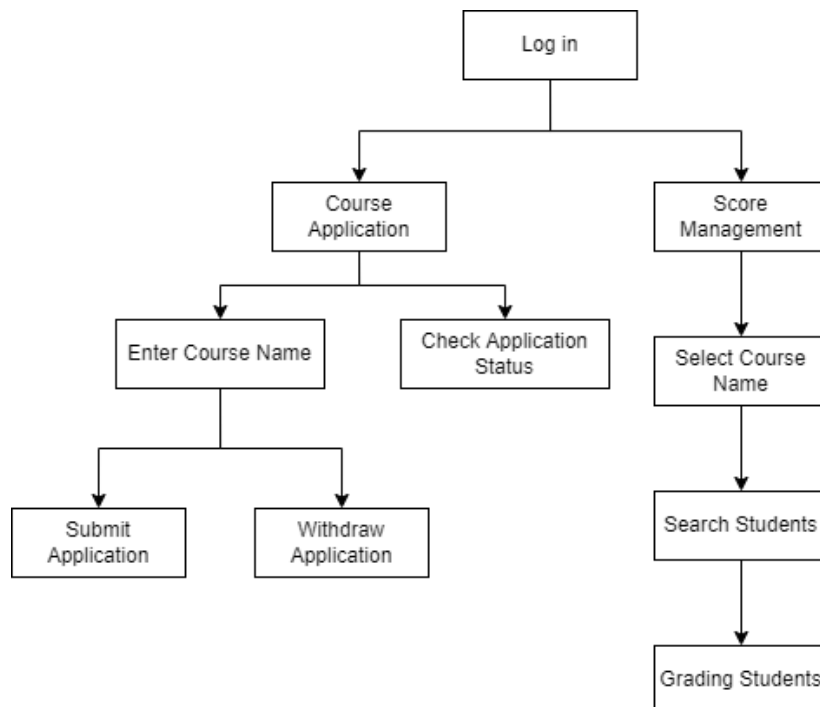


## 5. Final user flow

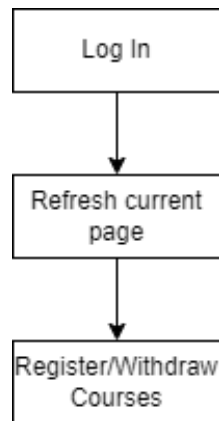
Administrator account:



Teacher account:



Student account:



## 6. Lessons learned

### (1) Technical expertise gained:

Through this project, we have a deeper understanding of object oriented programming (OOP). We use the “swing” package to design a Java GUI and we apply many tools in this package, like how to add a button, how to configure one section, etc. We need to design several classes for each kind of window by ourselves, which is really usable for Java programming.

Moreover, we add triggers when designing the database. We use triggers to make sure the user id will be created cascading with the teacher and student created. Through the EER model, we cannot find the relationship between user and teacher account because we do not add foreign keys, but the triggers can help them build the bridges.

### (2) Insights (time management insights and data domain insights)

We have three team numbers, but the project is also a huge task for us, so we need to manage time properly to finish the project in time. We give each teammate a subtask, like I design the management window, the other two guys design login window, teacher account window and student window. The logical design of database is comp up by us together. We search the material we need on the Internet and share the useful information with each other.

About the data domain, we take the “id” (teacher, student, course and college) to consideration for a long time. For the beginning, we want to use “INT” on the “id”, but after do some Java programming, we find that if “id” is “String”, it will provide more convenience for us.

### (3) Realized alternative design to the project

In this project, we want to use three types of users to constitute the course registration system. For the management, the “root” account can manage colleges, majors, teacher accounts, student accounts, and approve or reject course registration. For the teacher account, teachers can submit course registration to the manager, and they can score students’ courses. For the students, they can withdraw or register courses, and check the grades of courses they have registered.

For three types of users, the login window is the same. The program will confirm the user type via the constitution of input “id”.

(4) Document any code not working in this section

```
boxOneBox.add(Box.createHorizontalStrut(10));
```

We use this code to set the interval distance between each buttons or tables, but it seems that this code does not work for that we change the parameters but it does not make any change. We do not know the reason why it does not work but it might be one of the future works.

## 7. Future work

### (1) Planned uses of the database

We want to use the database to realize the course registration system. Because we already have designed a frontend interface, we can use this system in local environments. Our first thought is to mimic the Northeastern University course registration system to make a web application, but we find it is too difficult.

We still can find which course is the most popular to students through the student number for each course. Using this database, we can manage the colleges, majors, teachers and students, it is easy for the university to manage the education resource and make some adjustments.

### (2) Potential areas for added functionality

We want to add a student review system to give students rights for making comments on their courses. We can just link students' id to realize the system.

Also we want to find the relationships between teachers and students, like how many students a teacher once taught. We also want to track that the total courses a student has.

Mostly, we want to add a backend framework so that it can be in practical use. For java programming, we can use "synchronized" function to solve the "Multi-threaded concurrency" problem. We need to find a suitable framework for this project so that multi users can register courses at the same time. We can imagine the conditions that hundreds of students scramble to take a good course at the beginning of course registration.