

Academic Tracker Report

Kayla Nguyen, Rishabh Saxena

CS3200: Database Design

Kathleen Durant

June 21, 2023

## README

Ensure you have mySQL installed on your computer. If you do not have mySQL, here are steps to install: [mySQL Installation Guide](#)

Download the dump provided in the zip folder. If unable to access the file, access here:

[Academic Tracker Dump](#)

Import and run the files in mySQL: [Import Data Dump](#)

Download the application file: [Academic Tracker Application](#)

Ensure that your Java application is running JDBC

IntelliJ: [IntelliJ JDBC](#)

Eclipse: [Eclipse JDBC](#)

You can now run the application in the terminals of IntelliJ or Eclipse, or through another chosen terminal.

## Technical Specifications

This application is a database for users to keep track of their academic courses. It allows a centralized place, similar to a transcript, to see all of their classes. However, this application allows users to also keep track of people and resources they have accumulated from these classes. This makes resources and information more accessible to find, rather than having to look in multiple applications or notebooks for specific information. This database will allow users to filter what exactly they are looking for and where they can find where they stored their resources.

In this project, we utilized mySQL, in addition to code editors such as Eclipse and IntelliJ to write up the application code. mySQL was used in order to create the database and its stored

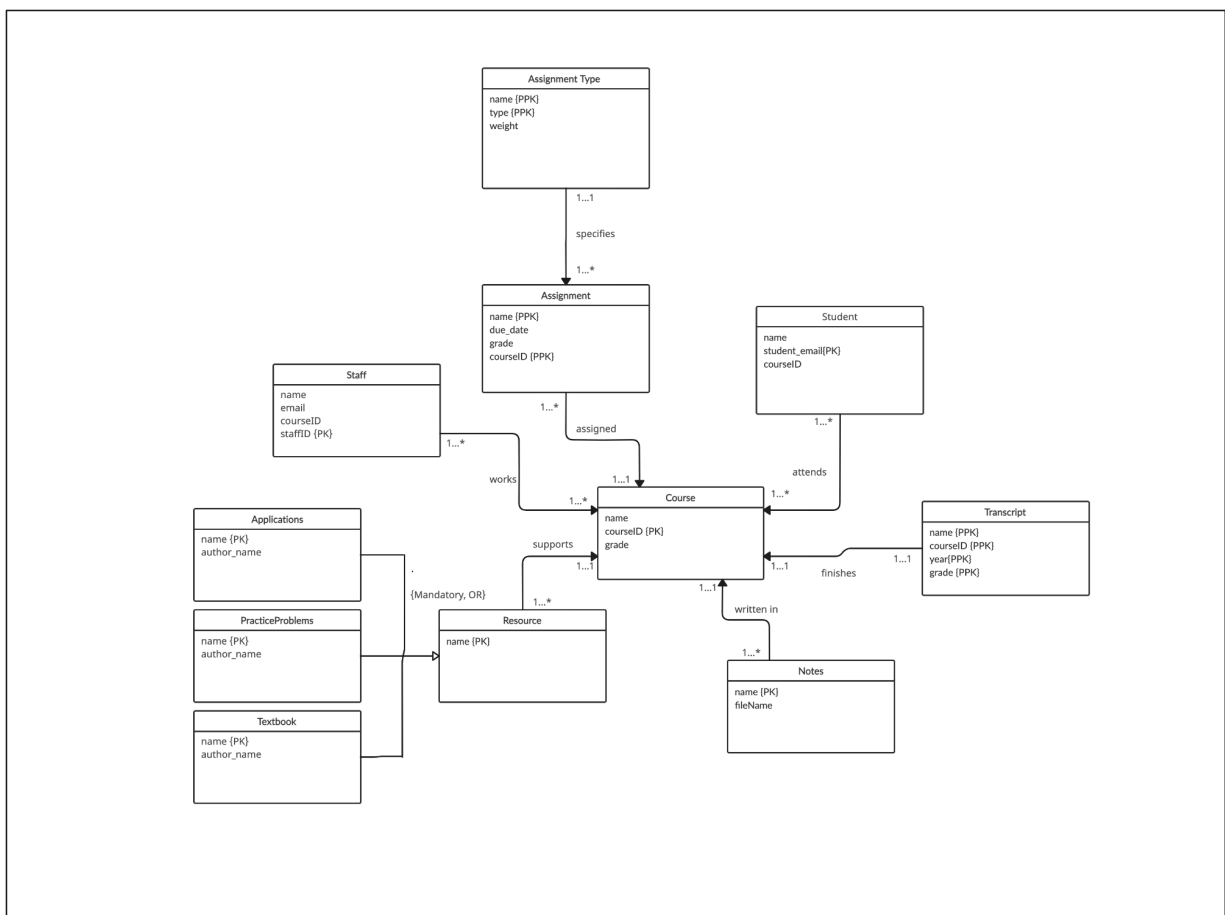
procedures that would be the backend of the Java application. In our Java code, we used the `javamysql` package in order to connect MySQL and the Java application. Additionally, the built in `Scanner` function of Java was used in order to ensure user interaction with the program through the terminal. Our program will be able to be run in the computer's terminal.

For our diagrams, we utilized MySQL and Creately. Creately was used to create the UML diagram and the user flow chart. We reverse-engineered our database in order to construct our EER diagram.

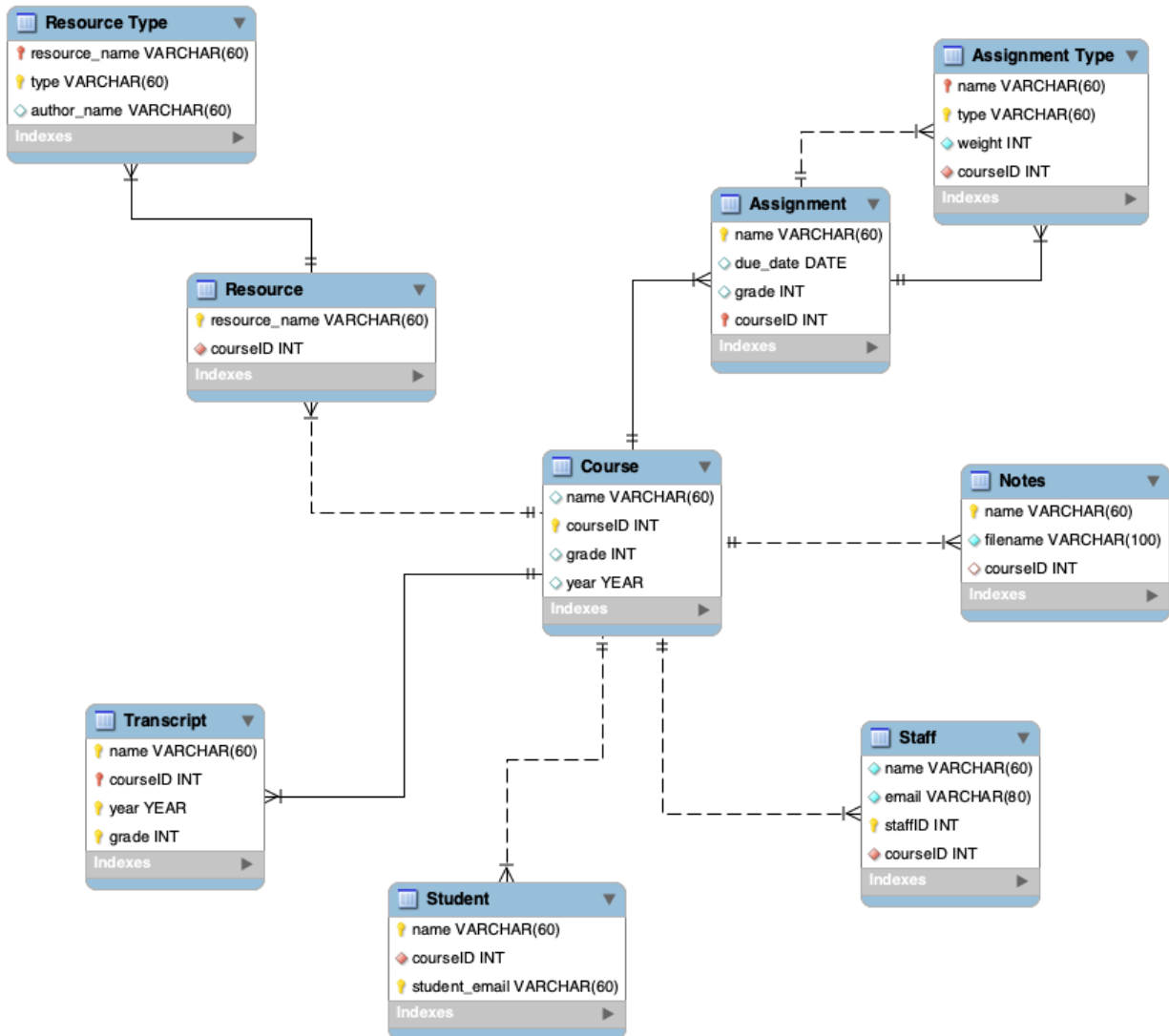
In the database, there are nine tables total – Course, Transcript, Notes, Resource, Resource Type, Staff, Student, Assignment, and Assignment Type. The Course table keeps track of the name of the course, the course ID, the grade the user currently has, and the year that they are taking the course. This entity is tailored more for current or future classes. The transcript table has the same fields as the course but serves more of a purpose if a person does not want to put any of their resources or it is a past course where they only have the information of the course. The notes table keeps track of the name of the note and where the note is located on the computer, which makes finding a note easier and efficient. The resource table keeps track of the resource name and the course the resource pertains to. The adjacent resource type table keeps track of the type of resource where we have specified three: an application, practice problems, or a textbook. In the resource type, it also specifies the author who wrote that specific resource. The staff entity keeps track of professors or teacher's assistants, storing their name, email, staff ID, and course ID. Most users probably do not know the staff's ID, but we assumed it may help in the future for indexing purposes, so the staff ID can be an arbitrary number of the user's choosing. The student entity is for students that the users may meet in their classes and may want to reach out to in the future, so it stores their name, the course they were in, and their student

email. The assignment table keeps track of the name of the assignment, its due date, the grade, courseID, type, and weight. The type and weight are stored in the Assignment table in order to satisfy third normal form. We have specified the type of assignments to be Homework, Project, Participation, Exam, and Other, which are the most common assignments to be graded, whereas Other will cover other assignments that do not fall into the category. As most assignments have a weight, the field stores the weight of each assignment type.

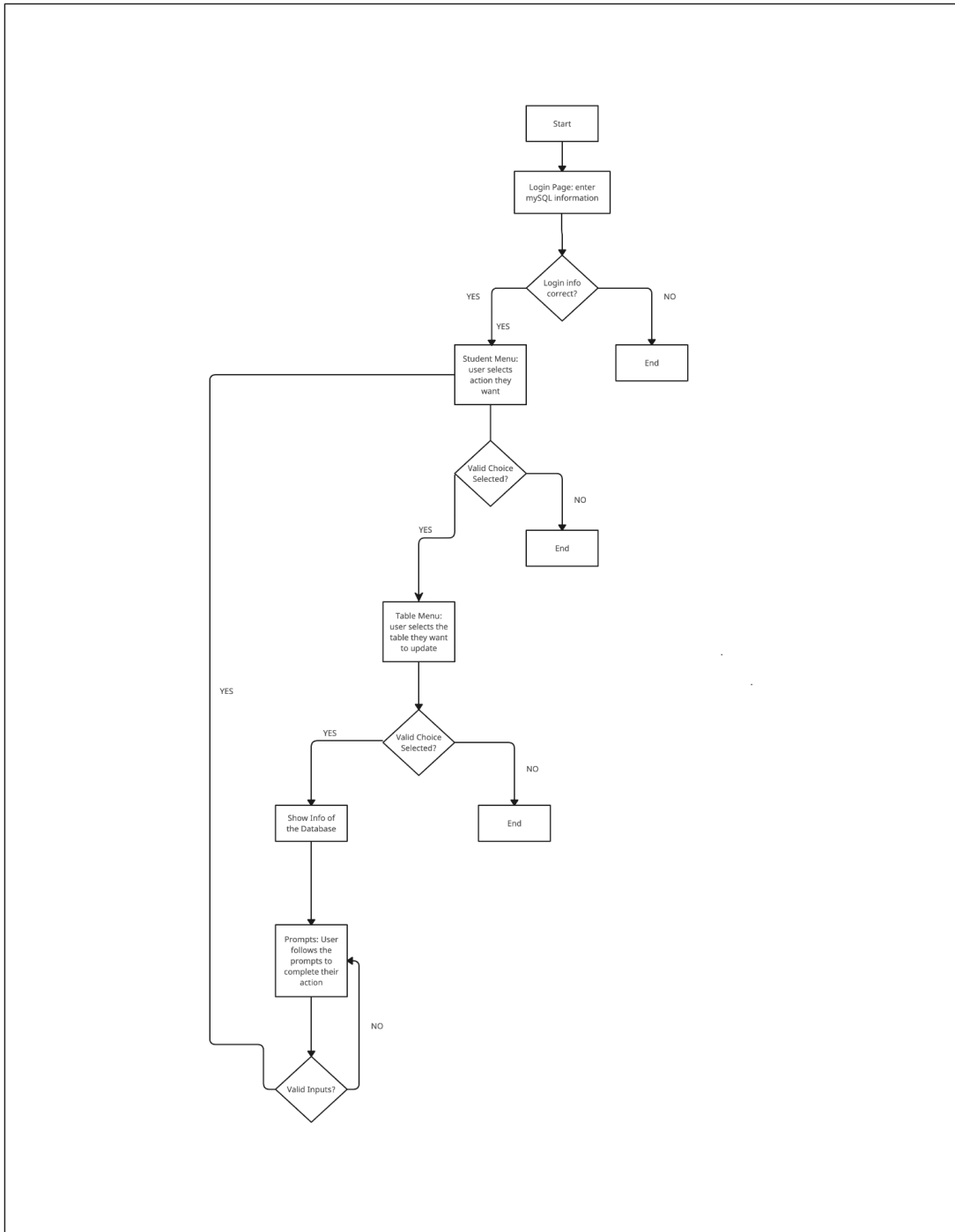
## UML



## Logical Design



## User Flow



## Lessons Learned

1. In this project, we learned how to better utilize MySQL and creating the database application. By utilizing the tools that we were given, we were able to become more familiar with MySQL, Java, and our prototyping skills such as utilizing Creately. We were able to practice creating stored procedures and creating a database, in addition to expanding our user interaction through the use of terminal menus. We practiced creating UML diagrams and user flow charts with Creately. We also practiced connecting Java to MySQL using `javamysql` as well as the scanner function of Java. We did have some difficulty as for the homework assignment, one of us did learned how to do it in Java and the other learned how to do it in Python. However, since both of us knew Java to begin with, the learning curve for connecting MySQL was easier than expected.
2. This assignment was particularly difficult with time management due to circumstances outside of our control. As full time summer students, we both were taking two classes in addition to one of us having work. This severely affected how much time we were able to spend on our project, but we were still able to manage to schedule time to meet and set goals. Our communication with each other could have been better in the beginning as we kept missing each other's calls, but we were able to pull through and were very communicative by the completion of our project. Additionally looking back, it would've been safer for backup of our database and information to learn how to store the database on Github, compared to what we were doing where we would make changes and send them through text. After creating the database, we found that our original design was not in third normal form. We fixed this by creating tables for resource and assignment types. The takeaway is to create databases originally in third normal form and keeping that in

mind, compared to an afterthought. We understand how to create and utilize databases in MySQL, but still need to work on creating databases with good design from the start.

3. When creating the database in mySQL, we found that parts of our original design wasn't intuitive enough for users. We were able to structure it so that it is course-based, where most of the information can be found through the courseID, keeping much of the information centralized to a certain class. We found this to be easier, as if we are the ones using it, we would want to see the notes, resources, etc, of our specific classes.
4. All code currently works.

### **Future Work**

1. We plan to use this database to keep track of classes that we have taken at our time at Northeastern. We will be able to keep track of courses, assignments, resources, students that have been in our classes, and professors that have taught us. This will not only be beneficial to allow us to access past and present resources, but we will also have information about students and professors that will allow for connections and communication in the future.
2. Some potential areas for added functionality is maybe making a letter grade aspect that people will be able to determine their GPAs for the semester, in addition to their cumulative gpa. Along the same lines, there could be a function that allows students to find their current grade where we take the weight and grade of the assignments in the course and find their grade. Another added functionality would be to specify the college that provides the course. We based it off just the course ID as we made the assumption that users would be able to just use the name of the course and the ID in order to identify the class. Adding the college might make accessibility of information easier: for example,



if they liked a course in a specific college and would like to take another course from that college. We mainly tailored this database to how we would use it as CS students, but maybe we could add more functionality so that it is more universal for other students. For example, our notes takes in a file path instead of a location, so there is an assumption that all of the notes are on the device that the user is using. As CS students, most to all of our notes are taken on that device. In future functionality, creating a location field would allow users to specify what device or notebook that their notes are located.