Project 4: Learning management System (Online quiz)

Kacey - Submitted Report

Chase - Submitted Vocareum Workspace

Instructions:

Compile by typing “javac MainMethod.java” into the terminal

Run by typing “java MainMethod” into the terminal

Description of classes:

**Teacher.java:**

When the teacher logs in with the correct username and password, they are provided with an interface. They have 8 different options. They can create a quiz, edit a quiz, delete a quiz, grade a quiz, view a student’s submission, create a new course, delete a course, and an exit option. If the teacher enters anything besides 1-8, they are prompted to enter a new number.

Almost every time the teacher is asked for input, input validation will occur. Meaning that if the teacher enters anything that doesn’t fit the options presented, they will receive an error and ask to try again. Or if the teacher enters a course, student, or quiz that does not exist, they will receive an error and ask to try again. Also, almost every prompt the teacher gets to enter something, they will also have the option to back out by pressing the letter that the prompt says. Most of the time it is N, but sometimes it is E.

When the teacher object is created, a new, empty file is created that will hold the courses the teacher creates. In order to actually get anything done with the teacher class, a course must be created first. When a course is successfully created, this empty file will now contain the new courses’ name. A new file will also be created which holds the quizzes within that course, which will be empty when first created. With a new course created, the teacher has several more options. They have two new options, creating a quiz and deleting a course.

If a teacher decides to delete a course, the course is removed from the text file which holds the teachers course. The text file that holds the quizzes in that course is deleted. Finally, all of the quizzes within that course are deleted. The student submissions are not deleted, however

If a teacher creates a quiz successfully, the previously empty text file that is supposed to contain the quizzes will now contain this quizzes name. A new file is also created when a quiz is made. This file is the actual quiz with the contents that the teacher provided. This file must be written in a certain way. The very first line will either contain true or false, which will determine if the teacher wants the quiz to be randomized. If there is no true or false, then it is assumed the quiz is already grade. The next line will be the question the quiz is asking. The next four lines are answer choices A, B, C, and D with their respective options. A new text file is also created with the correct answer choices that the teacher gives. While creating a file, the teacher is given an option if they want to import a file for the quiz. If they say yes, they must provide the file path to that file.. The quiz file must be formatted in the way that was just mentioned for it to work. The correct answers are asked for because an automated grading option was going to be implemented. However, we never got to that. So, when importing a file, no correct answers are needed.

With a quiz created, the teacher has several more options. The teacher can decide to edit the quiz. If this option is chosen, the entire quiz file is printed out with the line number before each line. The teacher can choose to edit a certain line or add another question. Remember that if the first line is edited to something not true or false, the quiz is said to be graded.

Another action the teacher can perform is the delete quiz option. If the teacher does this, the quiz file is deleted and the quiz name located in the course file that holds the quizzes is removed.

The last two options will only be performed correctly if a student has submitted a quiz. If a quiz has been submitted, a teacher can choose to view a student’s submission. This just prints out the students' submission of that quiz.

The last option is the ability to grade a quiz. If this option is chosen, the students' quiz will be printed one question at a time. The teacher can decide if the question is right or wrong. The grade is then calculated and written on the student submission file so the student can see their grade and what they missed. If the teacher tries to grade a quiz for a student who has not taken it yet, they will be returned to the options menu. Similarly, if no student has been added to this teacher’s course and the teacher selects one of these options, the teacher will be taken back to the options menu immediately as well.

One last thing to note about the teacher class. A space cannot be entered as the first character in any of the input. If a space is entered first, the space/spaces will be deleted and the rest of the characters will appear. So, technically a space can be entered first, but they will just be deleted.

**Student.java**

The Student class requires only the students username as a parameter so that it can later be used when writing the students file to pass back to the teacher for grading. The student class should be called after the login so that this parameter can be passed to it. The student class begins by prompting the user to input the username of their teacher. It will check if the teacher username they inputted exists by reading the TeacherAccounts.txt file to see if the username is stored there. If not it will print that the teacher does not exist. If it does exist then next the user will be prompted to choose a course out of the courses that the teacher has previously created. These courses are stored in a file named “*teachersUsername*\_Courses.txt”. If an invalid choice is inputted it will print an error and loop back to the course selection. Once a course is selected the student will be prompted to choose a quiz that the teacher has created. These quizzes are stored in a file called “*teachersUsername*\_*courseName*.txt”. Once a student selects a quiz the student class will check whether or not the student has taken the quiz before. This is done by checking for whether or not a file already exists called “*studentUsername\_teacherUsername\_courseName\_quizName.*txt”. This is the file that is created when a student completes a quiz. If the quiz has already been taken then the student will be prompted between returning to the course selection or viewing the grade on their quiz. If they choose to view their quiz then if the teacher has graded the quiz it will print the percent given by the teacher and if it has not been graded it will print that the quiz has not yet been graded. If the quiz has not been taken yet then the selection quiz will be printed into the terminal for the user to see. This is done by reading the quiz file and the printing with System.out.println(readLine()). The student will then have the option to import a file with answers or to answer manually. When the quiz is finished a file will be created called “*studentUsername\_teacherUsername\_courseName\_quizName.*txt” which will store a copy of the quiz taken, the students username, the students answers, and a timestamp of when they finished.

**Run.java**

For login, it will first print out a welcome prompt and ask if the user is a student or a teacher. If the user types in something that is not a 1 or a 2, the system will print out an error message and ask them to sign in again. After that, the system will print out a message to ask what action the user will like to proceed. Which includes making an account, editing an account, and sign in as an account. If the user decides to make an account, the system will prompt and ask them to enter their username, then the system will compare the username to the corresponding user type and see if the username already exists. If so, the system will return the user to the previous step. Or if the username contains a space, the system will deny the user from making an account. If the user decides to make an account and they pass all the requirements, the system will ask them for their password. Then the system will store the username in either teacherAccount.txt or studentAccount.txt based on their type, and the system will do the same for password. If the user already has an account and plans to edit their account, the system will first ask for their username, then compare it to the text file of their corresponding type. If the username does not exist, the system will prompt them back to the previous step. If the username exists, the system will ask for their password. Then the system will get the index of the account txt file and find the password in the password txt file using the index then compare it to the password the user typed in. If it matched, then the system will ask if the user wants to edit their username or their password, and using bufferedreader and bufferedwriter, the system will update the corresponding txt file. If the user decides to login, then similar to editing their account, the system will ask for their username, compare to the corresponding file to check if the username exist, then ask for password if the username exist, and compare the password the user typed in with what was stored before using the method mentioned above. If the password matches the username, then based on the type of the user, they’ll be able to use either functions from the teacher class or functions from the student class.