Final Project Interim Report

By

Jeffrey Trotz and William Varner

CT406

Web Programing Languages

Prepared for

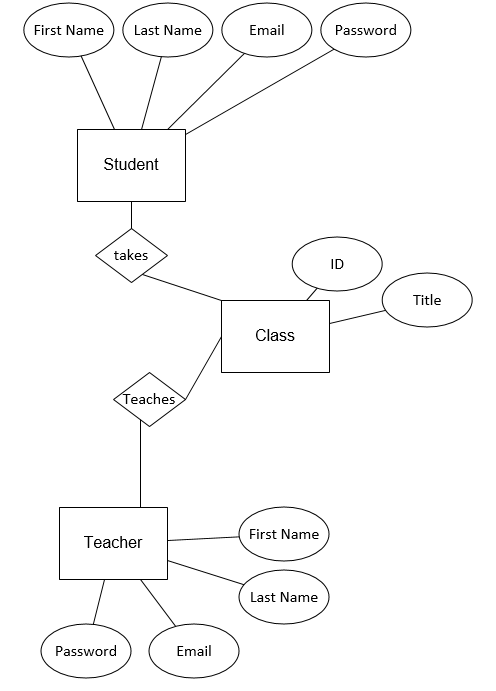
Dr. Eric Sabbah

Control Flow

The login page utilizes a servlet to log the student into their account. The servlet performs several checks before allowing the student to login: if the username or password fields are empty, it displays an error message and does not allow the student any further. If the username is valid, but the password is incorrect, it displays a different error message. If both fields are correct, then it logs the student into their account and forwards them to a page with a list of their classes.

The create account page also utilizes a servlet to verify the user’s input and direct them to corresponding pages. Like the login servlet, if any of the required fields are empty or if the username is already in use, it displays an error message. If all required information is present, the username is not taken, and the student ID does not already exist in the database, then it will create an account for the student and redirect them to a page with a list of their registered classes.

**ER Diagram and SQL to Create the Tables**



As shown in the ER diagram included, each student and teacher has eleven attributes that will be stored in the database: their student ID number, first/last name, password, email address, and classes. It is mandatory for the student to have one and only one student ID number, first/last name, username, password, and email address. I believe that it is possible to be registered as a student at a university, but not register for classes, so the student’s relationship to classes is zero or many. The student can create zero or one account because their student ID can only be used once to create an account.

The SQL below will create the student table:

CREATE TABLE STUDENT(

STUDENTID VARCHAR(10) NOT NULL,

FIRSTNAME VARCHAR(10) NOT NULL,

LASTNAME VARCHAR(10) NOT NULL,

PASSWORD VARCHAR(10) NOT NULL,

EMAIL VARCHAR(25) NOT NULL,

CLASSID1 VARCHAR(10),

CLASSID2 VARCHAR(10),

CLASSID3 VARCHAR(10),

CLASSID4 VARCHAR(10),

CLASSID5 VARCHAR(10),

CLASSID6 VARCHAR(10),

PRIMARY KEY(STUDENTID));

The SQL below will create the classes table:

CREATE TABLE CLASSES(

COURSEID VARCHAR(10) NOT NULL,

TITLE VARCHAR(25) NOT NULL,

STUDENTID VARCHAR(10) NOT NULL,

PRIMARY KEY(COURSEID));

The SQL below will create the teacher table:

CREATE TABLE TEACHER(

TEACHERID VARCHAR(10) NOT NULL,

FIRSTNAME VARCHAR(10) NOT NULL,

LASTNAME VARCHAR(10) NOT NULL,

PASSWORD VARCHAR(10) NOT NULL,

EMAIL VARCHAR(25) NOT NULL,

CLASSID1 VARCHAR(10),

CLASSID2 VARCHAR(10),

CLASSID3 VARCHAR(10),

CLASSID4 VARCHAR(10),

CLASSID5 VARCHAR(10),

CLASSID6 VARCHAR(10),

PRIMARY KEY(TEACHERID));

Test/Validation Plan

To test our LMS, we will begin by creating an account with a test student ID. Once the test account has been created, we will login to the account and view the list of classes that they are registered for, ensure that they can view and upload assignments, etc. We will also test error handling by logging in with invalid usernames or passwords and try to create multiple accounts with the same student ID.

The typical student and/or teacher should log in, see their classes spread in front of them, and then be able click on one of their classes to go to that classes page. The class’ page will include assignments, grades, documents, and a list of people in the class. The teacher will see these things but will be able to edit the grades section as well as upload documents.