**School Web App Project**

**Chris Leung**

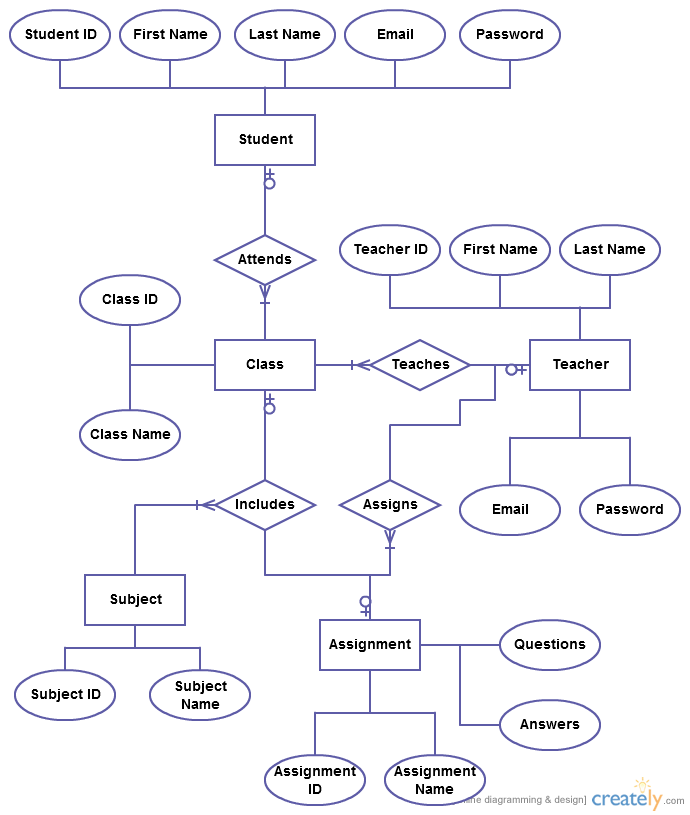
Requirements

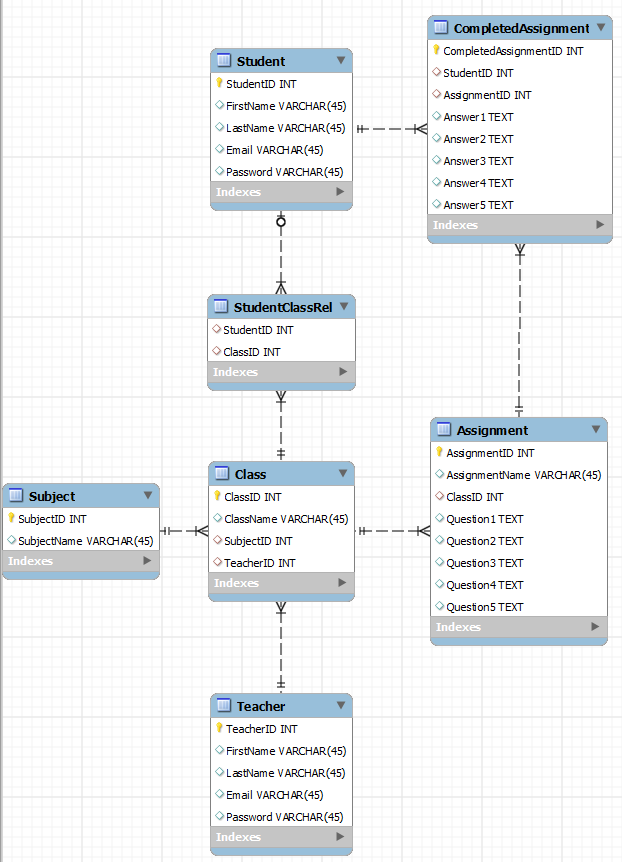
1. Students
   1. Can register and login
   2. Can see all their classes
   3. Can see what assignments each of their classes were assigned
   4. Can complete assignments
   5. Can turn in assignments
2. Teachers
   1. Can register and login
   2. Can see a list of their classes
   3. Can assign students to a class
   4. Can see which students in each of their classes
   5. Can create assignments
   6. Can assign assignments to classes
   7. Can check off assignments when turned in from students
3. Classes
   1. Have 1 teacher per class
   2. Have multiple students per class
   3. Can have 1 or more assignment per class
4. Assignments
   1. Can be created by teachers
   2. Are assigned to a class
   3. Can be filled out by students
   4. Can be checked off by a teacher

Database Tables

1. Students
   1. Student ID
   2. First Name
   3. Last Name
   4. Email
   5. Password
2. Teachers
   1. Teacher ID
   2. First Name
   3. Last Name
   4. Email
   5. Password
3. Class
   1. Class ID
   2. Class Name
4. Subject
   1. Subject ID
   2. Subject Name
5. Assignment
   1. Assignment ID
   2. Assignment Name
   3. Questions
   4. Answers
6. Student-Class
   1. Student ID
   2. Class ID
7. Teacher-Class
   1. Teacher ID
   2. Class ID
8. Assignment-Class
   1. Assignment ID
   2. Class ID
9. Subject-Class
   1. Subject ID
   2. Subject Name

ER Diagram





**General Notes for SchoolProject**

* All java servlet classes use doPost(): **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException
* Database: jdbc:mysql://localhost:3306/SchoolProject
* All inputs retrieved from request.getParameter( String name )

**index.jsp**

* Home page upon entering: <http://localhost:8080/SchoolProject/>
* Inputs: None
* Forms
  + Register
    - Inputs
      * Submit Button => RegisterServlet
    - Description: Click this register button to create a new account for this SchoolProject web app
  + Login
    - Inputs
      * Email text field
      * Password text field
      * Student/Teacher radio buttons
      * Submit button
    - Description: User enters their email and password to login to their student or teacher account. Will direct user to either Student or Teacher Home Page
* Description: Home page of SchoolProject to Register, create a new account for the web app, or login to their existing account

**RegisterServlet.java**

* URL: <http://localhost:8080/SchoolProject/RegisterServlet>
* Inputs: None
* Forms
  + CreateUser
    - Input
      * First Name text field
      * Last Name text field
      * Email text field
      * Password text field
      * Student/Teacher radio buttons
      * Submit button => CreateUserServlet
    - Description: Enter first name, last name, email, and password and select student or teacher to create a new user account for the school project web app
* Description: Register page to create new account and user in database in order to login to the school project web app

**CreateUserServlet.java**

* URL: <http://localhost:8080/SchoolProject/CreateUser>
* Inputs: first name, last name, email, password, student/teacher from RegisterServlet
* Forms: None
* Description: Displays message after a successful or unsuccessful creation of a student or teacher account and a link to the home page (index.jsp)

**LoginServlet.java**

* URL: <http://localhost:8080/SchoolProject/LoginServlet>
* Inputs: Email and Password
* Forms: None
* Description: Upon successful login, redirects user to either StudentHomePageServlet or TeacherHomePageServlet depending on user account type. Upon unsuccessful login, displays “Login Failed” message and a link to return to the homepage (index.jsp)

**TeacherHomeServlet.java**

* URL: <http://localhost:8080/SchoolProject/LoginServlet> or <http://localhost:8080/SchoolProject/TeacherHomePageServlet>
* Inputs: Email and Password
* Forms
  + CreateClass
    - Inputs
      * Class Name
      * Subject ID
      * Teacher ID
    - Description: Adds a class to the school with a name, subject, and teacher who teaches the class
  + ViewClass
    - Inputs: None
    - Description: Displays all the classes a teacher is teaching
  + CreateAssignment
    - Inputs
      * Class ID
      * Assignment Name
      * Questions 1-5
    - Description: Creates an assignment for a class with a name, and 5 questions
  + ViewAssignment
    - Input
      * Teacher ID
      * Class ID
      * Class Name
      * Subject Name
    - Description: View the assignments of the previously selected class
* Description: Teacher home page allows teachers to create classes, view the classes they are teaching, and create assignments for each of their classes

**CreateClassServlet.java**

* URL: <http://localhost:8080/SchoolProject/CreateClassServlet>
* Inputs: Class Name, Subject ID, Teacher ID
* Forms: None
* Description: Uses the inputs Class Name, Subject ID, and Teacher ID to create a class in the database for students to enroll in. Displays a message upon successful creation and a link back to the teacher home page

**CreateAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/CreateAssignmentServlet>
* Inputs: Assignment Name, Class ID, Question 1-5 (5 text areas)
* Forms: None
* Description: Uses the inputs to create an assignment for a class for students to complete.

**ViewAssignmentsServlet.java**

* URL: <http://localhost:8080/SchoolProject/ViewAssignmentsServlet>
* Inputs: Teacher ID, Class ID, Class Name, Subject Name
* Forms
  + ViewStudentsAssignments
  + Return to Teacher Home Page
* Description: Uses the Class ID, from the previously selected class, to display all assignments that belong to said class. User can click to view the students who have completed an assignment or return to the teacher home page

**ViewStudentsAssignmentsServlet.java**

* URL: <http://localhost:8080/SchoolProject/ViewStudentsAssignmentsServlet>
* Inputs: Student ID, Assignment ID
* Forms
  + GradeAssignment
  + Return to Teacher Home Page
* Description: Uses the Assignment ID to see which students have completed the assignment. User can click to grade a student’s assignment or return to the teacher home page

**GradeAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/GradeAssignmentServlet>
* Inputs: Completed Assignment ID, Questions 1-5, Answers 1-5
* Forms
  + CreateGradedAssignment
* Description: View an assignment’s questions and student’s answers. User can check off if the student answered correctly or not. If the assignment was previously graded, the previous score and checkboxes will be checked and shown. User can click to submit the grade to the database or return to the teacher home page

**CreateGradedAssignment.java**

* URL: <http://localhost:8080/SchoolProject/CreateGradedAssignment>
* Inputs: CompletedAssignment ID, CheckOff 1-5
* Forms: None
* Description: A graded assignment is created in the database with the inputs. User can return to teacher home page

**StudentHomeServlet.java**

* URL: <http://localhost:8080/SchoolProject/StudentHomeServlet>
* Inputs: Email and Password
* Forms
  + EnrollClass
  + View your Classes
  + CompleteAssignments
  + ListStudentsGradedAssignment
  + DropClass
* Description: The student home page. Students can enroll in classes, view their classes, view the assignments of each of their classes, and see if their assignments have been graded

**ViewCompleteAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/CompleteAssignmentServlet>
* Inputs: Student ID, Class ID, Class Name
* Forms
  + CreateCompletedAssignment
* Description: View the assignments of the previously selected class. Student can click to complete the assignment or return to the student home page

**AnswerAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/CreateCompletedAssignmentServlet>
* Inputs: Student ID, Assignment ID, Assignment Name, Questions 1-5
* Forms:
  + AnswerCompletedAssignment
* Description: Displays the questions of the previously selected assignment. Student can answer each question in a text area and click to submit and complete the assignment. If the assignment was previously completed, the previous answers will appear in the text areas and the student can change his/her answers

**CreateCompletedAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/AnswerCompletedAssignmentServlet>
* Inputs: Student ID, Assignment ID, Assignment Name, Answers 1-5
* Forms: None
* Description: Submits a Completed Assignment or updates a previous Completed Assignment in the database using the given inputs. Student can click to return to the student home page

**ListStudentsGradedAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/ListStudentsGradedAssignmentServlet>
* Inputs: Student ID, Class ID, Class Name
* Forms
  + ViewGradedAssignment
* Description: Displays the assignments of the class that was previously selected. Student can click to view the grade of the assignment or return to the student home page

**ViewGradedAssignmentServlet.java**

* URL: <http://localhost:8080/SchoolProject/ViewGradedAssignmentServlet>
* Inputs: Student ID, Class ID, Assignment ID, Assignment Name, Student First/Last Name, Questions 1-5, Answers 1-5, CheckOff 1-5
* Forms: None
* Description: View the questions, answers, and whether or not the answer is correct or not (as graded by the teacher) of the previously selected assignment. A score (in percentage) is displayed at the bottom or “Not Graded Yet” if the assignment has not been graded yet.S