QuizMaker Design Document

Group 15

Database Schema for quizdb:

user

user_id	email	account_name	password	role

class

class_id	prof_id	class_name	class_code

class_member

class_id	user_id

quiz

quiz_id	prof_id	quiz_name	possible_points	class_id
time_limit	reveal_answers	open_date	deadline	display_order

question

quiz_id	type	label	question_num
body	answer	graphic	points

mc

quiz_id	question_num	option_num	option_val	is_correct

fill_in

quiz_id	question_num	option_num	answer

matching

quiz_id	question_num	option_num	word	value

student_quiz

user_id	quiz_id	earned_points	graded	finished

student_question

user_id	quiz_id	question_num	student_answer
student_points	ta_comment	student_response	

student_mc

user_id	quiz_id	question_num	option_num

student_matching

user_id	quiz_id	question_num	option_num	answer

- The answer column in question and the student_answer column in student_question are only used if the question has one answer (true/false for the former, true/false, fill-in, and short answer for the later).
- mc, fill_in, matching, student_mc, and student_matching each represent a single option/answer in their respective question types. For student_mc, a row is inserted for every option the student checks.
- All columns with the same name across tables have a foreign key relationship. (The exception is the answer columns in student_matching and fill_in)

Design Schema:

1. General
    a. Sessions are maintained via a user_id session variable, which is retrieved on login/account creation, as well as a role session variable. The user_id and role combination is checked on every page to validate the user and check for permission to view that page. For example, students cannot view QuizMaker, and professors cannot view QuizTaker.

b. Pages containing forms post to controller php files that authenticate the user before parsing the form data and redirecting to the desired page, or redirect to the homepage if errors occur or unexpected input is received. All data is filtered using the php filter_var() method, and any data placed in the database is sanitized with mysqli_real_escape_string().

c. QuizMaker, QuizTaker, QuizGrader, QuizReview, all load quizzes via get requests.

d. Hidden inputs storing identifying information like class_code, account_name, etc. are often used throughout the application. Their info is sent along with the visible form data to the controller for use in authentication (for example, to check that the student actually belongs to the class).

2. Professor
    a. Creates and deletes classes
    b. Creates and deletes quizzes
    c. Grades quizzes
    d. Assigns and fires TAs
    e. Responsible for providing class code to students

3. TA
    a. Grades quizzes

4. Student
    a. Takes quizzes
    b. Reviews quizzes

5. Quiz Making
    a. Each quiz needs a quiz name, start time, and deadline.
    b. The start time and deadline are very important. Students in the class

can only take quizzes with start time before and deadline after the current time. Otherwise, they cannot even see the quiz in the quiz selection field.

c. Loading a quiz sends a get request, and the server directly prints the necessary html data into the page. No Json is used.
d. When a quiz is saved, it will immediately become available to students if the current time is within the start and end date. It is recommended that the date range be set in the past or far in the future until the quiz is actually fully created.

6. Quiz Taking
    a. Students must select a class and quiz from their user management page.
    b. Once the submit button is clicked, the student cannot retake the quiz.
    c. There is no progress save feature. Students must reload the quiz from scratch if they wish to continue at a later date.

7. Quiz Grading
    a. Both professors and TAs can grade quizzes.
    b. Select a class and quiz from the user management page. Then, in the grading page, select the student to load. Only students who submitted their quiz will show up in the selection field.
    c. Quizzes can be regraded.
    d. Student responses show up during grading. The fields will initially be empty, but will be filled when regrading if the student has responded.
    e. The save option saves the grading process and doesn't open the quiz for review to the student. The submit option saves the grading and allows the student to review the quiz.

8. Quiz Review
    a. Only students can review the quiz grade and make responses.
    b. Students must select a class and quiz from the user management page.
    c. Only quizzes that have been fully graded (the grader used the submit button) will appear in the quiz select field.

d. Quizzes can be reviewed multiple times.
e. The page that shows up in QuizReview is identical to the one in QuizGrader, but with the points and ta comments fields disabled and student response fields enabled.

9. Notes
    a. There is currently no error feedback for bad form input. On error, the controller will simply redirect the user without providing a description of the error.
    b. No php framework was used. All validation and database access is done manually.
    c. All validation is currently done on server-side.
    d. All quiz loads are done with regular get requests. No AJAX.