Proposed Requirements By Team

Software Design
SUNY Oswego
Spring 2017
Dr. Tenbergen and Dr. Harris

REQUIREMENTS

Nothing relevant was provided at the moment from the brainstorming exercise.

DATABASE

Store user, course and quiz information referred to in the other sections.

QUALITY ASSURANCE

The name for the system must not violate any laws.

USABILITY

Work with GUI to ensure that the interface is readable.

GUI

- Upon loading a new quiz, a countdown screen shall display to inform the user the quiz is about to start: ex *"Begin in 3... 2…"*.

- Each type of user shall be presented with a view of the system which makes sense for that user's capabilities.

- Answer statistics shall be able to be displayed via histograms, pie charts, etc.

- Users shall be given feedback after successful login E.g. *"Welcome to HCI 510!"*, *"You have successfully joined HCI 510!"*

- A homepage shall contain useful information for the user after login.

- The instructor shall be able to see what students are logged in to a given course/quiz.

- Upon successful answer/and or quiz submission, users shall receive feedback such as: *"Option A successfully submitted."*

- The time remaining for a given question/quiz shall be displayed during the course of the question/quiz.

- The interface shall be simple and intuitive, as well as aesthetically appealing.

- A selected but not yet submitted answer should be indicated to the user.

GUI CONTINUED

- Users shall receive positive feedback for doing well (Gamification).

- Upon error (*such as connection loss*) an informative message shall be displayed.

- When possible, error messages should offer a choice to the user to attempt fix the error Automatically (*reconnect button*).

- Created questions shall have one of several types of answers:

  ○ Multiple choice answers with one or more correct answers (*min-max answers ???*).
  ○ Extended response (*Text fields*).
  ○ Upload a file.
  ○ Slider Responses.

- If allowed by Instructor, students shall be able to move back and forth among the questions in a quiz, to change their answers.

- Confidence scale for questions.

- Instructors shall be able to post notes for a given course, visible to all Students in the course.

- Instructors shall be able to enter answers for a Student.

- Course class can be split into teams for trivia game quizzes. (*Do we want this? It sounds like it will require a bit more than it sounds like at first.*)

Languages to use (GUI):
HTML
CSS
JavaScript/Jquery mobile?
Bootstrap
Ajax?

Tools to use:
AMcharts (*for live, interactive charts*): https://www.amcharts.com/

Materialize: http://materializecss.com/

ENGINE

- System shall detect strange behavior that may indicate academic dishonesty (*ex: frequent logging in/out of multiple accounts during a quiz*).
- The system shall measure attendance for each quiz.

- Attendance information shall be available to the instructor of the course for a quiz.

- Instructors shall be able to input answers for their Students.

  Language (ENGINE): Groovy (*reasoning: seamless java integration, similar to Java*).
  Framework: Grails
  IDE: Intellij

- User Login.

  - A hash system shall be used for login security.
  - Users shall log in by a username and a password.

- Admins shall be able to assign courses to instructors (*Admin vs Instructor?*).

- Instructors shall be able to regulate which users are able to join that class/course/session.

- Instructors shall be able to edit their quizzes.

- Students' answers shall remain anonymous from other students.

- Students' answers shall be visible to the Instructor who owns the quiz. *(Instructor of the course*).

- While Students are connecting to a quiz, they should be informed via a message such as *"Waiting for quiz…"*.

- If a user submits an answer to a question, then submits a second answer, the second answer shall replace the first as the user's answer to the question.

  - On displayed question screen a counter shall display the number of users answers have been received from out of the number of users that can submit an answer (*ex: 5/13 shows that 5 users have submitted an answer, out of 13 users that are able to submit an answer*).

- After a quiz, a *"quiz review"* will appear displaying to the user their performance overall, and on individual questions. This review feature may be turned off, or restricted to overall results only by the instructor.

- The system shall have an API which will allow it to be used for various purposes not in the original design scope.

- Users shall be able to send private or public questions to the instructor. Private questions will be seen only by the sending user and the instructor. Public questions will be posted for all students to see. (*Other users can respond to public questions with comments?*)

- Open quizzes can created and stored by instructors under a tag (*ie: AST 100*) where other instructors may locate and use for their own classes.

- Instructors shall be able to search for a tag (*ie. AST 100*) to find open quizzes to use for their own classes.

- Users of different types will have different levels of access to the system's capabilities. Admins will have full control, while Instructors have control over their courses, and Students will be able to take quizzes as assigned by Instructors, ask questions, etc. (*This will become better defined as we negotiate and come to an agreement*).

- Instructors shall be able to create quizzes for their courses.

  - ○ A quiz editor shall allow an instructor to add new questions, edit existing questions, change the type of question (*the answer types*), change preset answer values (*multiple choice, for example*).

  - ○ The Instructor may set permissions for what abilities Students have during a quiz. This may include, for example, returning to previous questions.

  - ○ The Instructor may also select an evaluation type. Evaluation type will have to be discussed and defined. What is meant at the moment by evaluation type is how the Student shall find out how they did. This could be immediate (*after every question*), after the entire quiz, or after each question as a group, etc.

- Mobile and Desktop iterations ? (*Let's talk about this*).

- Instructors shall be able to export data

- The system shall receive submitted answers in under 7 seconds from the time a student submits an answer.

- Instructors shall be able to review a Student's quiz answers.

- Instructors shall be able to end a question timer early.

- Instructors shall be able to extend a question timer.