

Quiz Generator

The aim of this project is to automatically generate quizzes and to correct them at the end of each quiz. This will benefit to the students and to the professors as well. For the students this will help them to get trained for the exams, to test their knowledge in a specific course/domain, to put their selves in a real exam situation. They can also have their grades at the end of the quiz/exam instead of waiting for the professor's grades. For the professor this will be a huge gain in time and they can also build a data bases of questions that can be shared with other colleagues in other universities. Of course other uses of this software can be figured out (e.g., customer's satisfaction studies, on line tests, medical forms, etc.)

The customer (CS_UMass) wants quite ambitious software with useful functionalities. This document should be considered as a high level specification that draws the headlines of the project. So please, feel free to contact the customer for more clarifications. This can be in form of interview forms or by setting up face-to-face meetings with the customer.

Let's start with the first functionality, Quiz Generation. This can be done in 2 ways:

- A) Creating a quiz from scratch. The professor will have to fill-in a form in which she has to indicate :
 - The course to which this quiz belongs
 - The questions and the answers with the indication if it's a multiple choice question.
 - Marks allocated to each question
 - Notation strategy: zero or -0,5 if false
 - The difficulty level of each question (easy, medium, difficult, trap)
 - Estimated time to answer the question
 - Once the form validated, the quiz is generated for verification by the professor and then saved in the data base with a unique ID. Each question and its responses are also saved in the data base.

- B) Creating a quiz from the data base (of questions): each professor is in charge of creating a set of questions/answers for his course. Every question has a difficulty level, marks or weight on it, possible answers with a specific indication on the correct one (or many in case of multiple choice), the estimated time to answer the question (according to the prof) and the average time to answer the question (according to the historical exam records, this should be filled automatically and calculated after the exams). For statistical purposes, an indication of the success/failure ratio of the students on each question should be indicated.

Once the data base is populated with a set of questions/answers, professors can generate quizzes automatically. For that, they will need to go on a specific page/form, to indicate the number of questions, percentage of easy/medium/difficult/trap questions they want in their form, maybe they can also constraint the generation by either the average or estimated answer time of the quiz's questions. Once everything is set, a simple click will generate the quiz with a unique ID.

- C) Another important functionality is to create an exam event. This action consists in:
 - a. To precise a date, a room, time, number of students, and the name(s) of the professor(s) that have to watch the exam. Of course, the software should be able to make sure that the number of students fits the room's capacity, that the room is available and equipped with computers.

- b. To choose the quiz for the exam (either from the data base (already generated) or through a link to “create a new quiz”, or “generate a new quiz”
 - c. To choose a grading strategy (0 or -0,5 if the answer is false)
 - d. To activate the random order generation of the questions to avoid students cheating at the exam.
 - e. To precise the duration
 - f. To check the option if the students will have access to the results right away after the exam (grade and correct answers). In any case, the correct answers can't be displayed before the official end of the exam.
 - g. To define an access code to the exam event
 - h. Send the details of the exam event to all the students involved in the course
 - i. More other options can be figured out!
- D) Automatic grading of the Quiz: this is one of the main goals of this project! At the official end of the exam, each student may access to its results, grade (if allowed by the exam settings), can see his answers (good and bad). A color code should be defined (green questions answered ok, red no). An email also could be sent to all the students that participated to the exam with the grade and the correct answers. A file should be also generated with student's ID, name, and grade
- E) A student access should be of course granted to the student, to pass the exam, see its passed exams and grades, replay some tests, etc. The Student's ID is used to login!
- F) Professors should also have an access, to create exams, quizzes, questions, to look for an exam, for a student, etc.
- G) An admin is also needed to create accounts for the professors
- H) Professors should be able to add, update and delete courses
- I) Security is a super important aspect. To make sure that the students can't cheat during the exam. A solution should be figured out! Otherwise the software would be unusable and useless!
- J) The software should also provide monitoring facilities to the professor(s) for watching the exam.

Some possible extensions!

- 1) To be able to import quizzes or questions/answers in form of Word documents or Excel or Open Office, or any other format (up to you to propose)
- 2) A statistics tool to see for instance, to see the questions with the highest failure/success ratio, the most used questions, charts generated from the student's grades, the number of students that passes the course's exam, that fail, etc.
- 3) Super hard extension: to semantically check the quality of the quiz questions!! If the answer is not already in the question, to rate the question according to some best practices, etc.

Again, this is a high level description. Interviews with the customer may be required for a better understanding of the software needed.