MathTG – Progress Report
03/08/2016
Client: Luis A. Cueva-Parra

Developers:
Jason Hammond
Ernesto Contreras
Manal Abdalla

Our part in this project is to provide methods that facilitate the interaction with the database. The MathTG Contract document specifies two modes to generating a test: distinct selection of question and random generation mode.

General Structure

Before we start with the details of each mode, let us look at the general structure of the question bank. This sample table represents the organization of the database. Each course will have a list of chapters. Each chapter will have a list of difficulties. We will have three levels, from 1 – 3, where level1 is the easiest and level3 is the hardest. Each difficulty level will have a list of questions and answers in .html format.

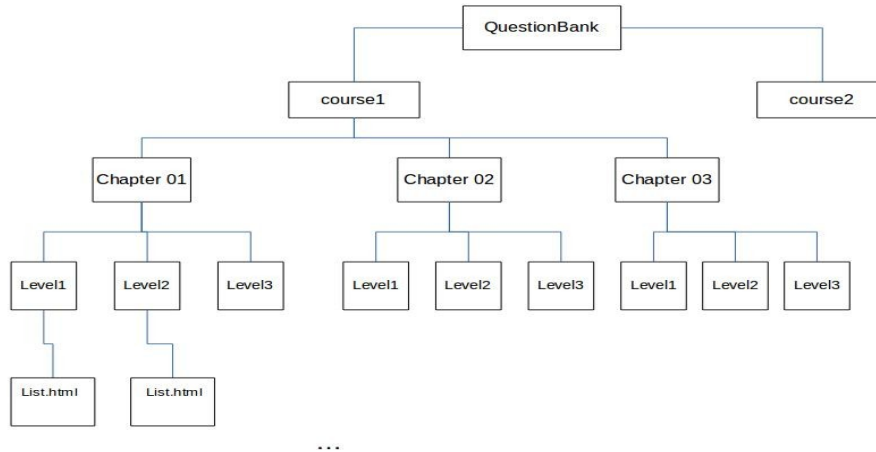| question bank										
Course math101										
Chapter 1			Chapter 2			Chapter 3			Chapter 4	
Difficulty 1	Difficulty 2	Difficulty 3	Difficulty 1	Difficulty 2	Difficulty 3	Difficulty 1	Difficulty 2	Difficulty 3	Difficulty 1	Difficulty 2
list.html	list.html	list.html	...	.....	.....	......	.....	......	......	....

MathTG – Progress Report
03/08/2016
Client: Luis A. Cueva-Parra
Developers:
Jason Hammond
Ernesto Contreras
Manal Abdalla

This structure is very simple and can be implemented using SQL-based database or simple directories like in the diagram above.

<u>Random Generation Method</u>
This method will return a list of file paths.
The parameters it takes:
1. Course name.
2. Chapter or chapters.
3. Type.
4. Total number of questions for the test and/or number of questions/chapter.
5. Optional: (number of questions from each level/chapter).

Notes:
- If number of questions from each level is not given (left zero), the test will be generated on the following statistics: 20% of total questions from level 3, 50% of total questions from level 2 and the remaining 30% from level 1.
- If number of questions per chapter is not given, the test will equally choose questions from each level between the chapters.
- The method will throw an exception if the number of questions requested from a level is out of range. It will also give the user the name of the course list that caused the error.

MathTG – Progress Report
03/08/2016
Client: Luis A. Cueva-Parra
Developers:
Jason Hammond
Ernesto Contreras
Manal Abdalla

- As output, the method will generate a .pdf file with the questions and their answers for the user's review. This method will not give a final format for the test form; it only provides the questions listed in the desired format.

Distinct Selection Method

This method will return a list of hand-selected file paths.
This method will populate the final list one chapter at a time.
It will take the following parameters:
1. Course name.
2. Chapter.
3. Type.
   At this point the method will display a .pdf list of all questions for the chosen chapter from the given course in another window.
4. The user will be prompted to enter the paths for the questions desired, Then, the user can move to another chapter if desired.
5. The final list will be displayed in .pdf format before being sent to the main method.
6. This method will also call the add and modify methods as needed.

Add Question Method

This method will allow the user to add questions to the question bank, specifying subject, difficulty level, and type. The method requires the following parameters:
1. Course name.
2. Chapter/Section.
3. Difficulty Level.
4. Type.

Once this information is received, the method will request first the question, then the answer. The method will then produce a new entry in the database, allowing the question to later be selected either directly or randomly using the appropriate methods.

Remove Question Method

This method will allow the user to remove questions from the bank. It requires only one parameter:
1. ID

This method will simply delete the question with the given ID.

MathTG – Progress Report
03/08/2016
Client: Luis A. Cueva-Parra
Developers:
Jason Hammond
Ernesto Contreras
Manal Abdalla

Edit Question Method

This method will allow the user to edit questions in case of discrepancy between question and answer. Only one parameter should be necessary to select the chosen question for editing:
1. ID

The method should then display the question and answer, and ask the user which he/she would like to change. Plain text input should be sufficient.

Browse Method

It is proposed that we also produce a method to allow the user to browse questions. This will require the following parameter:
1. Subject

This method will list all questions of a given method, along with their ID, Chapter/Section, and Difficulty. This will come in handy for the user that wishes to view the current state of the question bank, to retrieve IDs for deletion or editing.