# Cours Projet de Programmation

# Fiche Projet

## Project title : Multiple choice test generator and grader

## Student names

**Marnaoui Iheb Jouini Nesrine**

**Ramdhan Moez Khemakhem Wael**

## Functional Requirements Specification

*L’analyse fonctionnelle est une démarche qui consiste à rechercher et à caractériser les fonctions offertes par un produit pour satisfaire les besoins de son utilisateur.*

1. Who are the users (called *actors*) of your application? List the different types of users. (Typically there are only one or two types)

The application will have one actor only : The teacher.

1. How will each type of user interact with the application? (*user interface*)

* Our application will allow the user to generate a test containing a list of multiple choice questions.
* Our application will allow the user to choose the number of questions in the generated test.
* Our application will allow the user to grade a list of tests contained in a given directory.
* Our application will allow the user to grade a given test (the user will be asked to enter the path of the file containing the test and the answer key) [based on a given test ID specified by the user].
* Our application will allow the user to load a set of new multiple choice questions from a formatted file to the database.
* Our application will allow the user to add a new multiple choice question to the database of questions.
* Our application will allow the user to access with a login and a password.
* Our application will allow the user to view a previously generated test.
* Our application will allow the user to print a multiple choice test.

## Program Design (Conception Détaillée)

*Il s’agit de la décomposition de votre projet en des tâches et des sou-tâches, pour une meilleure gestion du projet. Concevoir donc une classe pour chaque tâche. Les méthodes de chaque classe correspond à des sou-tâches, et les attributs correspond au données gérées par cette tâche.*

For each class, you should give the following information below :

* name of the class
* the task that this class represents
* all methods and what they represent
* all data members and what they represent

class 1

* Name of the class : Question
* The task that this class represents :

This class represents a question.

* All the methods and what they represent :

question() : default constructor

Question(char str[]): parameterized constructor

Question(char str[],int a)

addQuestion() : allows the user to add a question to the text file

static addQuestions(string filePath) : allows the user to load a set of new multiple choice questions from a formatted file.

toString() : return a string that represents a question

getQuestionId() : return the Id of a question

getQuestionId() : will allow the user to creat a question manually

getCorrectAnswer() : return the correct answer of the question

getTotalNumberQuestions() : return the number of questions saved in the database

* All data members and what they represent :

long int questionID : question identifier

string description : Represents the statement of the question.

string [ ] choices : Represents the list of choices.

Char   correctAnswers : Represents the list of correct answers.

class 2

* Name of the class : Test
* The task that this class represents

This class represents a group of questions.

* All methods and what they represent :

Test(int numberQuestions): parameterized constructor.

Private generateTest(int numberQuestions): generate a new test

static printTest(long int testID): print a generated test.

static checkTest(int testID): consult a test.

* All data members and what they represent :

long int testID : identifier of the test.

list < int>  QuestionsID : identifier of the questions.

int numberQuestions: number of questions.

TestAnswers answers: object that represents the correct answers.

class 3

-        Name of the class : Answers

-        The task that this class represents :

This class is an abstract class that represents the answers to a test.

-        All methods and what they represent :

Answers (long int answersID): parameterized Constructor.

Answers (string file) : parameterized Constructor

~Answers() :destructor

* All data members and what they represent

int answersID: Identifier of an answer

map<long int IdQuestion , string [ ] correctAnswers> answerdetails: a map that represents a question (by its identifier) and the corresponding correct answers.

class 4

* Name of the class : TestAnswers extends Answers class
* The task that this class represents

The answers to a generated test.

* All methods and what they represent

testAnswers (string file) : parameterized constructor.

testAnswers (long int testAnswersID,map<long int IdQuestion , string  []correctAnswer> map ) : parameterized constructor.

gradeTest(StudentAnswers s): allows the user to grade a given test.

Private generateTests(string Dirpath) : allows the user to generate the answers file to a test

~TestAnswers() : destructor

generateAnswers() : will allow the user to generate answers

grade(StudentAnswers S) : will allow the user to grade tests

* All data members and what they represent : no data members.

class 5

* Name of the class : StudentAnswers extends Answers class.
* The task that this class represents :

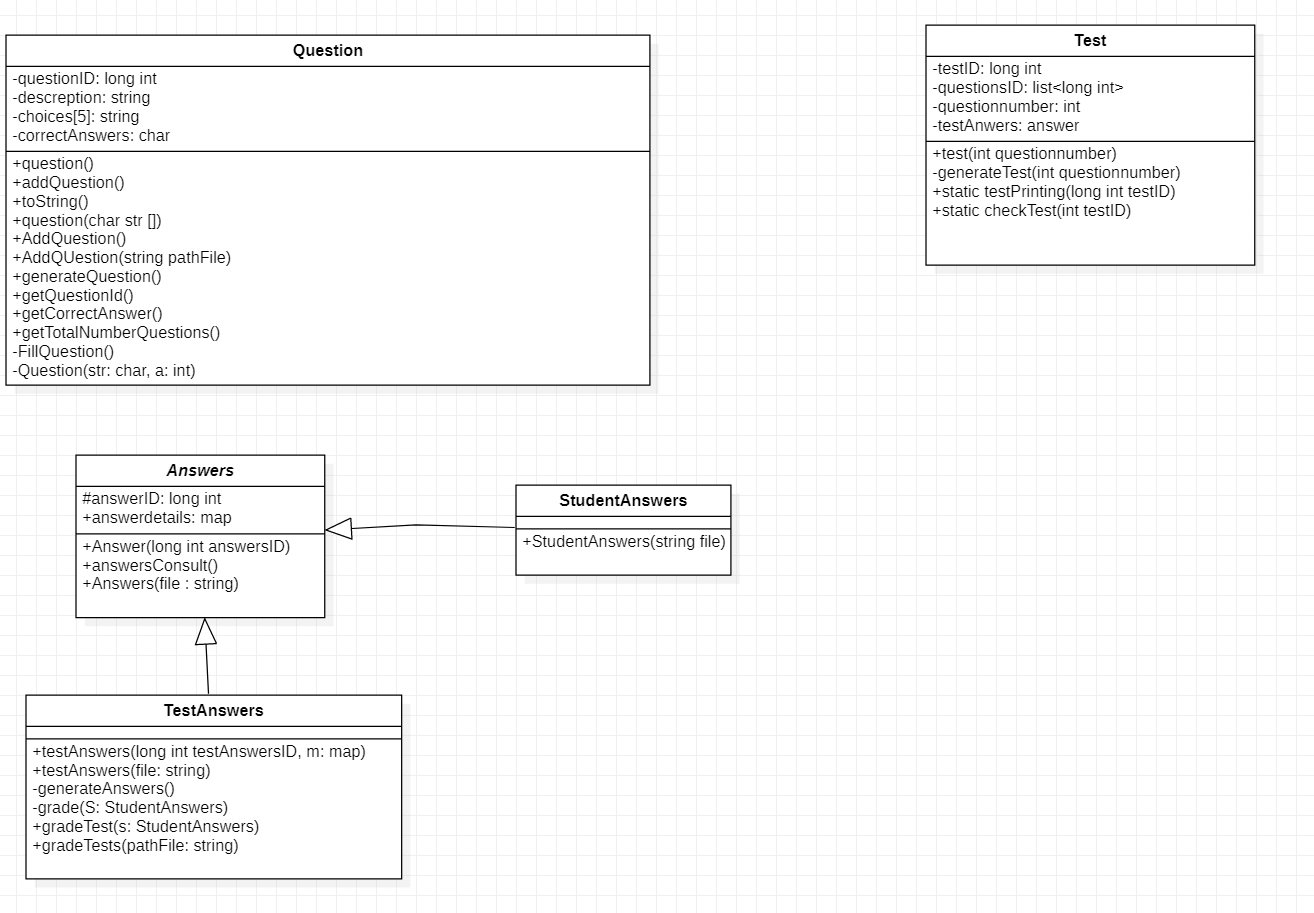
Represents a student answers to a given test that are yet to be corrected.

* All methods and what they represent :

StudentAnswers (string file): a file path containing a student’s answers to a given test.

~StudentAnswers()

-All data members and what they represent : no data members

Class Diagram