Vincent Alves
Pierre Merlin d'Estreux de Beaugrenier
Thibault Gimenez
Benjamin Legrand

Projet Génie Logiciel 2: Livrable 1

31 mai 2014

Table des matières

1 Introduction

Ce projet Génie Logiciel 2 que l'on nous a assigné doit aboutir à la réalisation d'un gestionnaire de QCM.
Le but du projet est de développer un logiciel de gestion et de traitement de QCM : Questionnaires à choix multiples. Ce logiciel est destiné à une école et a pour but d'évaluer les élèves sur des modules en leur faisant passer des QCM. Il a également pour but de donner le ressenti des élèves à travers des QCM sur les modules.
L'équipe en charge de ce projet est formée de Pierre Merlin d'Estreux de Beaugrenier, Thibault Gimenez, Benjamin Legrand et Vincent Alves.

Afin de former l'équipe en charge de ce projet, nous avons décidé de conserver l'équipe du premier projet Génie Logiciel et d'y rajouter un quatrième membre (Vincent Alves).

Dans ce document, nous décrirons les besoins du client vis-à-vis de ce projet tels que nous les avons compris, et détaillerons notre planning prévisionnel pour les mois à venir.

2 Reformulation des besoins

En se basant sur les demandes du client, nous avons établi les listes de besoins suivantes :

2.1 Besoins fonctionnels

- Trois types d'utilisateurs : L'administrateur, les professeurs, et les élèves.
  - Créer des niveaux d'accès différents pour les trois types d'utilisateur : L'administrateur peut accéder à plus de ressources que le professeur, qui peut accéder à plus de ressources que l'élève.
- L'administrateur peut définir les utilisateurs et les promotions. Les promotions sont des listes d'élèves. Les élèves sont définis par leur nom et prénom.

- L'administrateur peut définir les modules enseignés à l'école par leur nom, la liste des modules prérequis et le syllabus du module.

- Un professeur peut créer des QCM, qui sont un ensemble de questions avec un libellé accompagné d'une liste non vide de réponses fermées.
  1. Les QCM définis peuvent être privés(utilisable par le professeur uniquement) ou publics(utilisable par tous les professeurs)
  2. Chaque réponse d'un QCM est définie par un libellé et une information précisant si elle est vraie ou fausse.

- Un professeur peut créer des sessions de QCM. Une session est définie par ses dates de début/fin (attribut de la session) ainsi que le module et la promotion auxquelles elle est associée.
  1. Un professeur qui a créé une session de QCM peut consulter à tout moment les résultats partiels(si elle n'est pas terminée) ou définitifs de cette session. Il peut voir les résultats de chaque élève et les statistiques.
  2. Les résultats demandés par le professeur peuvent être les scores individuels des élèves participants, ou des statistiques sur l'ensemble des élèves : Moyenne, écart-type, fréquence de bonnes réponses par question.

- On doit pouvoir connaître la liste des modules qu'enseigne un professeur.

- Un élève peut répondre à un QCM s'il est inscrit à la session et si la session est ouverte.
  1. Un élève peut refaire une session de QCM tant qu'il n'a pas atteint un nombre de répétitions précisé par le professeur.
  2. Un élève peut consulter les résultats d'une session à laquelle il a participé, à la condition qu'elle se soit terminée.

- Le client nous impose l'utilisation de JAVA 1.7.

2.2 Besoins non-fonctionnels

- Le logiciel doit avoir un temps de réponse inférieur à 10 secondes pour toute action.

- Le logiciel doit être facilement utilisable par des étudiants et professeurs n'ayant pas nécessairement de connaissances avancées en informatique.

- La réalisation du logiciel doit être divisée en plusieurs jalons.
- De ce fait, la communication avec le client doit être fréquente à tous les jalons pour vérifier que la réalisation va dans la direction souhaitée par le client.

- Le logiciel doit pouvoir fonctionner sous Windows ou GNU/Linux.

3 Application de la méthode SIXO

Pour aider à la réalisation de notre planning, nous avons appliqué la méthode SIXO :

3.1 Objectifs

Créer un logiciel de gestion et de traitement de questionnaires utilisable par les professeurs et les élèves et géré par un administrateur.

3.2 Objet

On peut distinguer trois objets :
- L'analyse (reformulation des besoins, méthode SIXO et planification du projet)
- La conception (création des diagrammes avant de coder son projet et des statistiques utile pour le traitement des questionnaires)
- La réalisation (Le code java et jeu de test)

3.3 Opérations

Il faut reformuler les besoins, analyser le projet avec la méthode SIXO, planifier le projet, créer un UML, faire des diagrammes de classes, d'activités, de séquence, d'états-transitions, créer des statistiques et coder le projet avec le langage XML.

3.4 Ordre

1. Reformuler les besoins
2. Analyser le projet avec la méthode SIXO
3. Planifier le projet
4. Réaliser les diagrammes UML : Classe, Activité, Séquence, Etat transition.
5. Réalisation technique du projet en utilisant le language java
6. Tests du projet fini pour vérifier la satisfaction des besoins.

3.5 Opérateurs

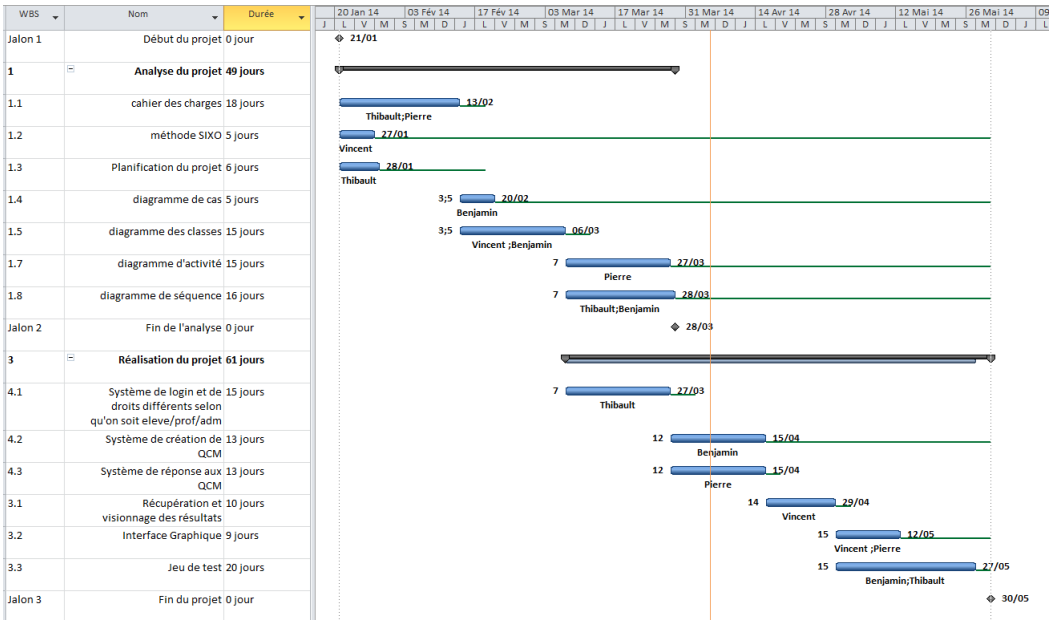
Quatre étudiants : Pierre Merlin d'Estreux de Beaugrenier, Thibault Gimenez, Benjamin Legrand et Vincent Alves.
Pour l'analyse 2 binômes sont utiles. Un pour l'analyse des besoins et la méthode SIXO, l'autre pour la planification du projet et l'UML. Pour la conception, il faut un binôme pour les statistiques et un pour les diagrammes.
Pour la réalisation, les quatre étudiants travaillerons sur le codage ensemble et selon leur compétences.

3.6 Outils

- Language Java version 1.7
- Language UML
- StarUML
- Ordinateurs
- Conseils des professeurs
- Outils suite Office(Excel, Word, Project)
- LaTeX

4 Planning Prévisionnel

Il a été choisi par l'équipe de diviser le projet en deux livrables. Une concernant l'analyse du projet et l'autre concernant la réalisation du projet. Ce choix est dû, du faite que de nombreux membre du projet sont listeux. Il aurait été donc difficile de rendre de nombreux livrables. On a ainsi le planning suivant :



La partie programmation débute dès que le diagramme de classe a été réalisé. Les zones en vert représentent les marges. Pour plus de détails le diagramme de Grant est disponible en pièce jointe.

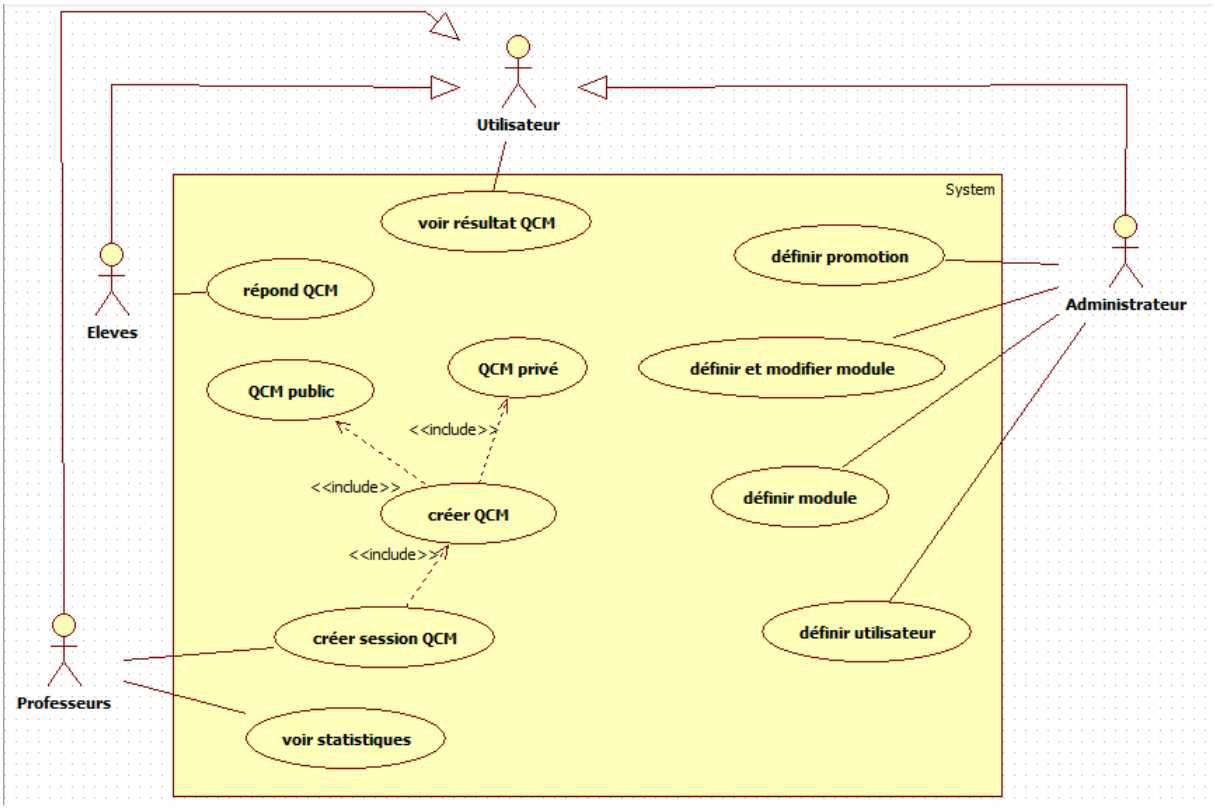
5 Analyse du projet

5.1 Diagramme de cas

Le diagramme de cas sert à identifier :

- Le système
- Les acteurs qui interagissent avec le système
- Les actions des acteurs sur le système

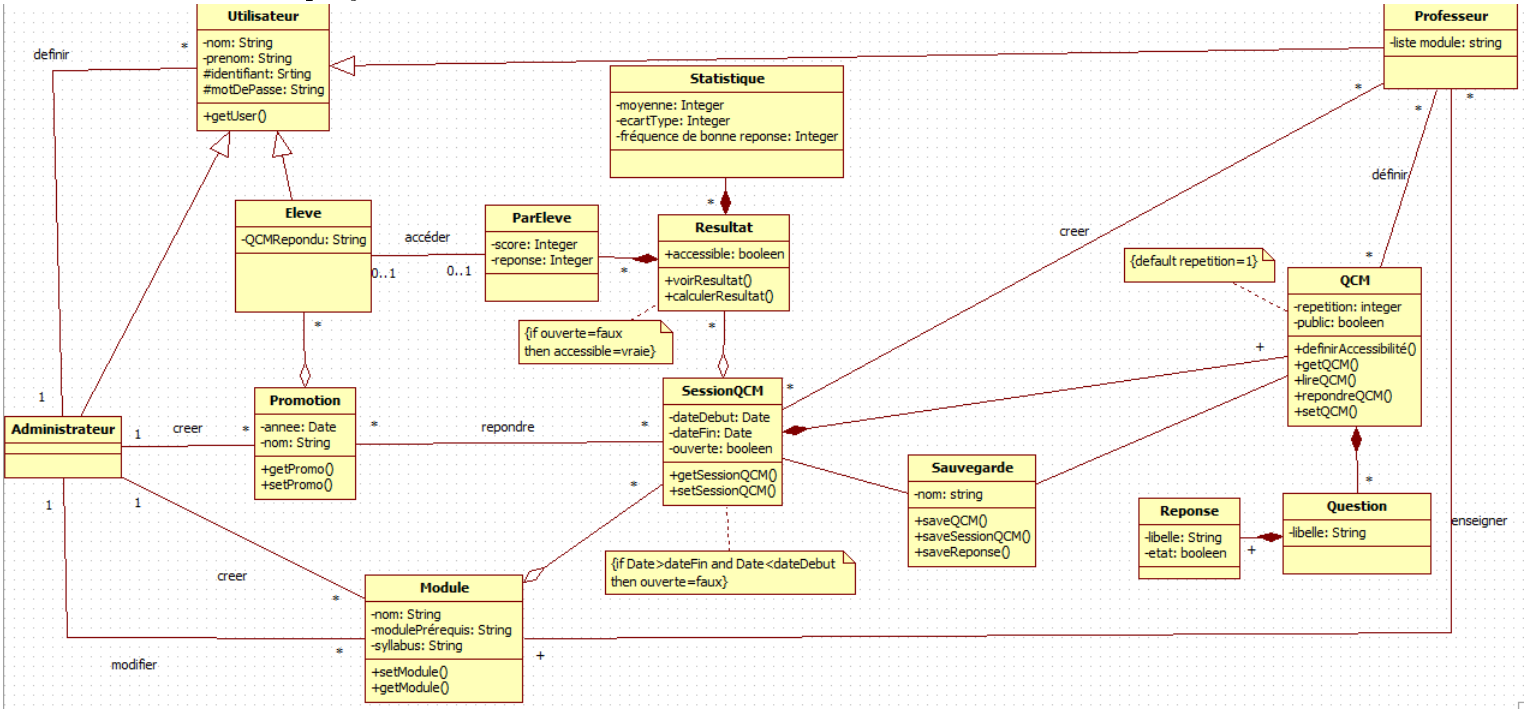
Ce qui nous a permis d'établir le diagramme de cas d'utilisation suivant :



Un acteur est une entité externe (personne, machine...) qui interagit avec le système. On distingue ainsi trois types d'utilisateur (administrateur, professeur et élève) avec des actions différentes. Tous les détails des actions des utilisateurs ne sont pas représentés pour une question de lisibilité.

5.2 Diagramme de classe

Le diagramme de classe est le diagramme le plus indispensable à la programmation en Java. Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Le diagramme de classe du projet est le suivant :



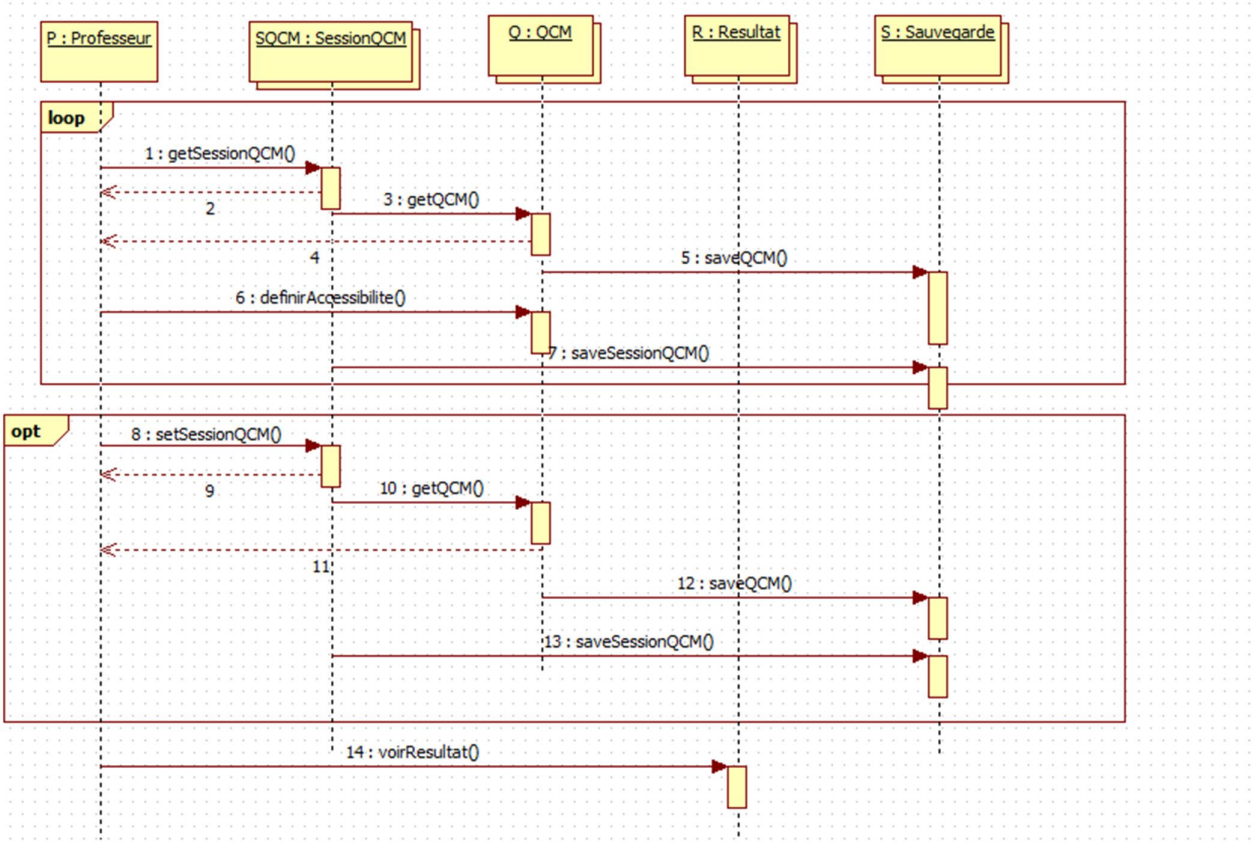
Le diagramme de classe est divisé en trois parties. L'une comprenant les utilisateurs (professeur, élève et administrateur). Une autre comprenant les QCM, les modules et les promotions. C'est-à-dire les données créées par l'administrateur et les professeurs. La dernière partie du diagramme de classe contient les résultats des QCM.

5.3 Diagramme d'activité

Voir le fichier joint « projet5 » pour les détails de ce diagramme. Il a été choisi de faire des boucles pour mettre en évidence le fait que l'utilisateur puisse effectuer une nouvelle action après avoir effectué une action. Des sauvegardes ont été rajoutées avant la déconnexion de l'utilisateur afin de sauvegarder les modifications de l'utilisateur. Nous avons choisi une boucle pour la connexion avec trois essai de saisie de mot de passe afin d'assurer la sécurité des données du logiciel.

5.4 Diagramme de séquence

Ci dessous, le diagramme de séquence pour le professeur.



Pour les autres diagrammes de séquence voir la pièce jointe « projet5 ».
Le diagramme de séquence donne l'ordre de l'action effectuée par l'utilisateur. Il a été choisi de mettre des retours après la création et la modification des QCM pour indiquer à l'utilisateur que les modifications ont été effectuées.

6 Conclusion

Ce premier livrable contient donc toute la partie analyse du projet qui a été réalisé par un travail en groupe. Il nous reste donc toute la partie implémentation en java à réaliser. Nous avons pris un léger retard au niveau du codage mais cela peut se rattraper pendant les vacances en particulier. Pour ce travail en équipe il n'y a pas encore eu de problème de communication ou interne au groupe comme des mésententes entre personnes du groupe.