Cahier des charges

1. ***Equipe en charge du projet***

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 procéder à la formation de l’équipe en charge de ce projet, nous avons décidé de garder l’équipe du projet GL1 et de rajouter un quatrième membre (Vincent Alves).

1. ***Reformulation des besoins***
2. ***Besoins fonctionnels***

En se basant sur les requêtes du client, nous avons défini la liste de besoins client suivants :

* Il existe trois types d’utilisateurs : L’administrateur, les professeurs, et les élèves.
* Le client nous impose l’utilisation de JAVA 1.7.
* 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 également définir les modules enseignés à l’école par leur nom, la liste des modules prérequis et le syllabus du module.
* Un professeur qui a créé une session d'un QCM, peut consulter à tout moment les résultats partiels ou définitifs de cette session. Il peut voir les résultats de chaque élève et les statistiques.
* On doit pouvoir connaître la liste des modules qu’enseigne un professeur.
* Un professeur peut créer des QCM, qui sont un ensemble de questions avec un libellé accompagné d’une liste non vide de réponses fermées.
* Un QCM est un ensemble de questions avec un libellé, une liste de réponses fermées. Chaque réponse est définie par un libellé et une information précisant si elle est vraie ou fausse.
* Un professeur doit pouvoir définir des QCM qui sont soit privés (utilisable par lui uniquement), soit publics (utilisable par tous les professeurs).
* Un professeur peut créer des sessions de QCM. Une session est définie par ses dates de début/fin ainsi que le module et la promotion auxquelles elle est associée.
* Une session de QCM est définie par une date de début, une date de fin, est associée à un module, à une promotion et possède un nombre indiquant une répétition (par défaut qui vaut 1).
* Un élève peut refaire une session de QCM tant qu’il n’a pas atteint le nombre de répétitions mentionné précédemment.
* Un élève peut répondre à un QCM s’il est inscrit à la session et si la session est ouverte.
* Un professeur peut consulter à tout moment les résultats d’une session qu’il a créée, même si elle n’est pas encore terminée.
* Un élève peut consulter les résultats d’une session à laquelle il a participé, à la condition qu’elle se soit terminée.
* 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.

1. ***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 personnes n’ayant pas de connaissances avancées en informatique.
* Diviser la réalisation du logiciel en plusieurs livrables
* Communiquer fréquemment avec le client (le responsable du projet).
* Le logiciel, réalisé sous java, doit pouvoir fonctionner sous windows ou linux.

1. ***Méthode SIXO***

1. Objectifs

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

1. 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, XML, jeu de test)

1. 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 les langages java et XML.

1. Ordre

1 : reformuler les besoins

2 : analyser le projet avec la méthode SIXO

3 : planifier le projet

4 : Le UML

5 : Les diagrammes de classe, d’activité, de séquence et d’état transition

6 : créer des statistiques

7 : coder le projet avec les langages java et XML et faire une interface

8 : Le jeu de test

1. 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.

1. Outils

* Langage java version 1.7
* Langage UML
* StarUML
* Ordinateur
* Conseils des professeurs
* Excel
* Document Word
* Microsoft projet