Génie Logiciel 2 : Cahier des charges

Christian Ingouff
Pierre-Alexandre Tyndal
Sonia Seddiki
Yann Charbonnier

EISTI

2013/2014
Semestre 2

Table des matières

Partie 1

Présentation générale du projet

1.1 Contexte et objectifs du projet

L'objectif principal de ce projet Génie Logiciel 2 est la réalisation d'un projet informatique dans la continuité directe du sujet du premier semestre, qui était en l'occurrence « Gestion de questionnaires ».

La réalisation d'un projet informatique se doit de respecter les contraintes imposées et d'être pertinent dans son utilité. Cette dernière nous demande de développer un logiciel de gestion et de traitement de questionnaires à choix multiples (QCM). Ce logiciel est destiné à une école et a pour but d'évaluer des élèves sur des modules en leur faisant passer ces QCM.

Les objectifs à remplir au cours du projet sont donc :

- Définir le(s) administrateur(s)
- Définir les professeurs, créateurs de QCM
- Définir des élèves (qui seront entre autres les interviewés)
- Définir la hiérarchie des droits des utilisateurs en fonction de leur rôle
- Mémoriser les QCM entrés
- Mémoriser les réponses des interviewés aux différentes sessions de questionnaires
- Réaliser le programme en langage « Java »
- Avoir un rendu ergonomique (look, facilité d'utilisation, ...)
- Fournir une documentation exhaustive et claire

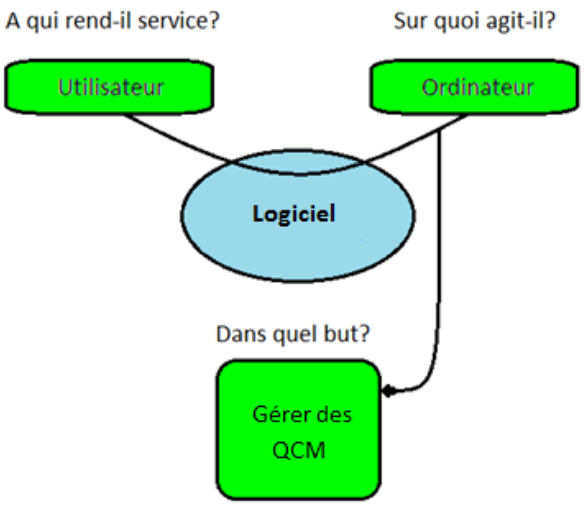

FIGURE 1.1 – Utilisateurs concernés et enjeu : Bête à corne

1.2 Acteurs du projet

Ce projet est réalisé par un groupe de quatre acteurs :

- INGOUFF Christian
  - Contact : ingouffchr@eisti.eu
  - Connaissances : Pascal, C, OCaml, HTML/CSS/PHP, SQL/gestion BDD, VBA
- TYNDAL Pierre-Alexandre
  - Contact : tyndalpier@eisti.eu
  - Connaissances : Pascal, C, Java, OCaml, HTML/CSS/PHP, SQL/gestion BDD
- SEDDIKI Sonia
  - Contact : seddikison@eisti.eu
  - Connaissances : C, Maple, SQL/gestion BDD
- CHARBONNIER Yann
  - Contact : charbonnie@eisti.eu
  - Connaissances : C, SQL/gestion BDD

1.3 Intérêts pédagogiques

Ce projet est, comme pour le premier semestre, une occasion d'organiser et gérer un projet grâce à des méthodes telles que la méthode « SIXO » ou le diagramme de Gantt via Microsoft Project. Ce travail, effectué en groupe, requerra également des compétences abordées dans le module « Travail en équipe ».

D'un point de vue technique, nous développerons nos connaissances dans le langage de programmation « Java » et nous organiserons notre travail sous le principe de la programmation objet et du « pattern MVC ». Cette rigueur nous donne une approche de la programmation au niveau professionnel.

Partie 2

Expression fonctionnelle des besoins

2.1 Environnement

Le but du logiciel est de gérer des QCM. Nous avons donc besoin de lister les éléments qui constituent l'environnement et créent les contraintes imposées à sa réalisation. Ces éléments sont les suivants :

- L'ordinateur
- L'espace mémoire disponible
- Les normes et législations
- L'utilisateur

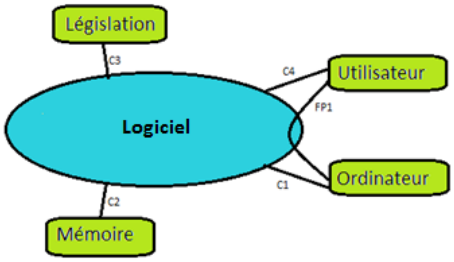

FIGURE 2.1 – Diagramme pieuvre des contraintes

2.2 Description des fonctions et contraintes

L'ensemble des éléments constituants l'environnement impose un certain nombre de contraintes :

Contraintes	Intitulé
C1	Pouvoir interagir avec l'interface
C2	Être le moins spacieux possible
C3	Respecter les normes et législations imposées par le sujet
C4	Être esthétique et ergonomique

D'autres contraintes nous sont aussi imposées :

- Langage de développement : Java version 1.7
- Persistance des données : Sérialisation des objets proposée par Java
- Cette technique de sérialisation doit être masquée par une interface (au sens UML) afin de pouvoir changer simplement de méthode de persistance de données
- Analyse et conception en UML
- Utilisation du pattern MVC
- Dans un souci de maintenance et d'évolution possible, les objets doivent communiquer entre eux à travers des interfaces
- La sauvegarde des données se fait automatiquement à la sortie du programme ou à chaque fois que l'utilisateur le demande. Seules les données modifiées, crées ou supprimées doivent être mises à jour lors d'une sauvegarde.

Un ensemble de fonctions contraintes devront permettre de réaliser l'application :

- Un administrateur définit :
  - Les utilisateurs
  - Les promotions (listes d'élèves)
  - Les modules, dont il peut définir ou modifier
    - Le nom du module
    - La liste des modules pré-requis
    - Le syllabus du module

- Un utilisateur est défini par :
  - Son nom
  - Son prénom
  - Son rôle : enseignant (préciser liste de modules), élève ou administrateur

- Un professeur peut :
  - Définir des QCM (privés ou publics) : liste de questions
    - Si un QCM est privé, seul le professeur qui l'a créé peut l'utiliser. Sinon, tous les professeurs peuvent l'utiliser.
    - Les questions ont un libellé et présentent une liste de réponses
    - Les réponses ont un libellé et un attribut vrai ou faux selon leur véracité
  - Définir des sessions de QCM :
    - Date de début
    - Date de fin
    - Module associé
    - Promotion associée
    - Nombre maximum de fois pour répondre au QCM
  - Consulter une session de QCM s'il l'a définie :
    - Réponses et score final de chaque élève
    - Statistiques sur l'ensemble des élèves : moyenne, écart-type, fréquence de bonnes réponses par question

- Un élève peut :
  - Répondre à une session de QCM à laquelle il est inscrit, entre la date de début et la date de fin de session
  - Visualiser ses résultats à une session de QCM à laquelle il a participé, à la suite de la session

Des fonctions facultatives complèteront les fonctions principales décrites précédemment :

- Authentification : Tout utilisateur devra s'authentifier afin de pouvoir utiliser le logiciel
- Interface graphique : Esthétique et utilisation aisée
- Interaction inter-utilisateur : Messagerie entre les différents utilisateurs

Partie 3

Ressources

3.1 Ressources humaines

Nous disposons de personnes présentes pour nous superviser et nous aider dans notre projet. Ces dernières sont :

- Superviseur informatique : Mme. NGUYEN Nga
- Superviseur RH : M. ABDELMOULA Nejib

3.2 Ressources techniques

L'ensemble des ressources techniques se divise en deux parties :

- Les ressources techniques mises à disposition
- Les ressources techniques à acquérir

Les ressources techniques mises à disposition sont :

- LaTeX : professionnalisme pour les rapports
- Langage Java : réalisation du programme
- Eclipse : IDE pour la programmation en Java

Les ressources techniques à acquérir sont :

- Méthode d'analyse à l'aide de diagrammes UML
- Connaissances sur le langage de programmation Java

3.3 Ressources temporelles

Maintenant que le projet est visualisé, nous avons établi grâce à Microsoft Project une organisation temporelle du travail, afin de hiérarchiser les différentes étapes du projet. Le diagramme est également disponible en image plus nette dans le fichier *gantt.png* fourni.

FIGURE 3.1 – Diagramme de Gantt du projet

Partie 4

Structuration SIXO

Après avoir établi les principales lignes de notre projet, nous pouvons conclure en l'organisant selon la méthode SIXO, en répondant aux six points :

Objectif	Gestion de QCM
Objets	Cahier des charges, Analyse UML, Conception, Programmation Java
Opérations	Utilisation d'outils et de ressources (MS Project, UML, Java)
Ordre	Structuration en 5 modules successifs
Opérateurs	Une école
Outils	Répartition des opérations en rôles, gestion de l'équipe