

# Plateforme pour apprentissage et auto-évaluation

Projet du module “Génie logiciel avancé”

Année 2013-2014

## 1 Le but du projet

Ce projet est une étude de cas concrète pour la conduite d’un projet informatique. Son but principal est de vous mettre dans la situation d’un projet assez important, qui doit être effectué en équipe et en respectant les contraintes de spécification et implémentation vues dans le cours de “Génie logiciel avancé”.

Par conséquent, l’accent sera mis sur la méthode de gestion de l’équipe du projet, sur la spécification du projet (cahier de charges, modules), sur son architecture logiciel, sur son interopérabilité avec d’autres projets et sur sa fiabilité. Peu de contraintes seront imposées sur la partie implémentation : vous avez la liberté de choisir l’architecture de votre logiciel ainsi que le langage de programmation.

## 2 Le sujet

### 2.1 Contexte

Le réseau “FUN – France Université Numérique”, la plateforme française de cours en ligne, a ouvert le 16 janvier 2014. Plusieurs cours numériques sont ouverts pour les 100 000 inscrits de la plateforme. Des plateformes similaires sont disponibles aux Etats Unis depuis 2010 et ils affichent un nombre d’inscrits de plus de 6 millions [2].

Un module important de ces plateformes est le système de validation des connaissances acquises dans un cours pour la cohorte d’étudiants inscrits. Nous vous proposons de spécifier, développer et tester une telle plateforme numérique pour la validation de connaissances.

### 2.2 Cahier de charge informel

Dans la suite, nous vous donnons une *description informelle* des fonctionnalités du produit logiciel que vous devez développer. Cette description est volontairement incomplète ou avec des incohérences car elle doit être formalisée dans le cahier de charges de votre projet.

**Devoirs :** Le suivi d’une cohorte d’inscrits à un cours est organisé selon le modèle suivant :

- Des feuilles (pages) d’exercices sont associées à chaque chapitre du cours (un chapitre prend en général une à deux heures de cours). L’étudiant doit effectuer ces exercices et les rendre sur la plateforme, ce qui correspond à des TD/TP notés. Une semaine de cours peut comprendre plusieurs chapitres afin d’atteindre un volume de 6 heures de travail personnel par semaine et par cours.
- Les chapitres sont groupés en parties et chaque partie est associée à des feuilles d’exercices de révision. Ces révisions peuvent prendre la forme d’un rendu de mémoire ou d’un projet.

— L'examen final d'un cours est constitué d'un ensemble d'exercices qui doivent être rendus avant la fin du cours. Le sujet de l'examen final peut être disponible dès le début du cours. Plusieurs formes d'exercices doivent être supportés par la plateforme :

- des *questions à choix multiple* (QCM),
- des *questions à réponse formatée* (QRF) où la réponse doit être donnée dans un format bien précis. Par exemple, à la question

“Quelles sont les étapes de l'algorithme de tris par insertion sur le tableau d'entiers [1, 4, 2, 3, 5] ?”

la réponse attendue est une séquence de tableaux (séparés par une nouvelle ligne) dans le format utilisé par la question,

- des questions dont la réponse est un programme (P) dont les entrées et les sorties doivent respecter un format fixé par l'exercice,
- des questions dont la réponse est libre (L) mais limitée, en général, par un nombre maximum de mots.

**Evaluation :** Chaque rendu est validé par un nombre de points. Plusieurs types de corrections peuvent être effectuées en fonction du type d'exercice :

- une correction automatique est appliquée pour les exercices de type QCM, QRF et P ;
- la correction par pairs consiste à faire corriger une réponse par un autre étudiant inscrit au cours ;
- une correction par l'enseignant, surtout pour les questions à réponse libre (L) et pour évaluer le style de programmation dans les exercices de type programme (P).

Un étudiant inscrit peut soumettre un nombre fixé de fois un devoir avant la date limite. Le dernier score obtenu est conservé. Le score d'un exercice est affiché ou transmis par courriel immédiatement après la soumission quand une correction automatique est possible. La plateforme assure le lien (anonyme) avec le correcteur d'un exercice.

**Interface avec l'étudiant :** Un étudiant inscrit sur la plateforme peut suivre plusieurs cours en même temps. A son connexion sur la plateforme, un tableau de bord est affiché avec les devoirs qu'il doit accomplir et leur date limite.

Il peut commencer ou continuer à travailler sur un devoir et puis (re)soumettre ce devoir ou le conserver sans le soumettre.

A la (re)soumission d'un devoir, le nombre de points obtenu est affiché si le devoir est corrigé automatiquement, ce qui sert aussi d'acquiescement. Pour les exercices sans correction automatique, la soumission doit être acquittée par un message courriel. L'arrivée d'une correction doit être notifiée sur le tableau de bord.

L'étudiant est également notifié quand une correction lui est attribuée. Il doit pouvoir consulter de manière anonyme la réponse et soumettre sa note avec une justification.

**Interface avec les professeurs :** Les professeurs sont responsables de la création de feuilles d'exercices dans le format accepté par la plateforme. Pour chaque exercice d'un devoir, il doit préciser le barème. De plus, il doit préciser la correction à utiliser. Par exemple :

- pour un exercice QCM, chaque choix de réponse doit avoir un nombre de points ;
- pour un exercice QRL, la réponse attendue doit être indiquée ;
- pour un exercice P, le professeur doit fournir un ensemble de fichiers incluant un fichier de type `makefile` avec deux entrées pour la compilation du programme et son test ; les autres fichiers peuvent être des libraires ou des fichiers utilisés pour la compilation et le test,

— pour un exercice L, le type de correcteur (pair ou professeur) doit être fixé.  
La configuration d'un devoir peut être faite sur plusieurs connexions du professeur, jusqu'à ce qu'il décide de rendre publique le devoir.

Le tableau de bord du professeur doit indiquer pour chaque cours qu'il encadre l'état d'avancement des devoirs et le travail de correction qui lui est attribué. Le professeur peut également obtenir pour chaque cours des statistiques sur le taux de réussite des exercices, des devoirs et de la cohorte en général.

### 3 L'organisation du projet

Pour ce projet, nous vous imposons une organisation bien précise du travail :

**Conduite du projet :** Vous devez d'appliquer une méthode de gestion "agile" avec 3 itérations (*sprint* dans la langue de la méthode agile) de 3 semaines (environ) :

**1ère itération :** du 3 février au 26 février,

**2ème itération :** du 4 mars au 25 mars,

**3ème itération :** du 2 avril au 6 mai.

Ainsi, vous devez rendre 2 versions provisoires (pour les premières deux itérations) de votre projet avant la soutenance finale.

Pour gérer les différentes versions de votre logiciel, vous devez utiliser un gestionnaire de versions et donner accès aux professeurs du cours à votre archive. Une option pour le gestionnaire est le serveur `git` de l'UFR.

**Organisation des équipes :** Les équipes seront constituées de 5 à 6 étudiants. Chaque équipe doit désigner un responsable de produit (*product owner* – celui qui décide quelles fonctionnalités seront implémentées) et un responsable d'itération (*sprint manager* – celui qui organise le travail). Les responsables ne doivent pas "seulement" définir le travail à réaliser et distribuer les tâches à leurs co-équipiers mais aussi ils doivent s'assigner des tâches. Attention, chaque membre d'une équipe sera noté selon sa participation au projet, les responsables du projet doivent veiller à une répartition équitable des tâches.

Chaque groupe doit également désigner un responsable pour l'interface avec les autres groupes (voir paragraphe ci-dessous *Inter-opérabilité des projets*).

**Contenu des rendus :** Après chaque itération, vous devez fournir un rendu du projet et réaliser une démonstration aux horaires qui vous seront fixés en TD. Les rendus doivent contenir :

- le cahier de charges général et le cahier de charge de l'itération,
- de la spécification fonctionnelle,
- de la spécification architecturale,
- de la spécification d'interface de chaque module du projet,
- du plan de test et de validation,
- du code implémenté et des résultats des tests,
- le journal de bord de l'itération constitué de compte-rendus hebdomadaires.

Les spécifications des interfaces du module définition d'exercice et du module correction d'un exercice doivent être faites en utilisant les types abstraits de données. Les autres spécifications ainsi que le cahier de charges utiliseront les diagrammes UML.

Ces rendus doivent être disponibles sur votre dépôt `git` et bien identifiées (avec un tag) comme étant des rendus de fin d'itération.

**Inter-operabilité des projets :** Les équipes peuvent communiquer pour échanger leur expérience sur les modules du projet. L'échange de code ou de spécification/documentation est interdit.

Tous les groupes doivent adopter le même format pour la spécification (et le stockage) de devoirs et de leur correction. Pour cela, des séances de TD seront dédiées à un travail inter-équipes afin de définir ce format. Vous pouvez également utiliser pour cela la liste de diffusion du projet `m1gl-projet@informatique.univ-paris-diderot.fr`. Le but du format commun est de pouvoir échanger les devoirs entre les plateformes.

Vous êtes également encouragés à réaliser des modules qui sont inter-opérables avec les modules des autres projets. Il est ainsi plus facile d'obtenir un produit logiciel très performant en ciblant uniquement le développement d'un module de votre produit.

**Extensions :** Des idées d'autres fonctionnalités qui sont utiles pour ce produit peuvent être trouvées en s'inscrivant aux plateformes de cours numériques cités ci-dessus ou en consultant le document [1], disponible sur l'ENT.

## 4 L'évaluation

La note finale du projet sera calculée en utilisant les éléments suivants :

- le rendu de chaque itération,
- le document final incluant le cahier de charges, la spécification et le plan de test du projet, et
- la présentation orale du projet.

La soutenance du projet aura lieu après la session d'examens. Chaque groupe aura 30 minutes maximum. Nous vous recommandons d'utiliser un support visuel (planches) et de faire intervenir chaque membre de l'équipe.

## Références

- [1] Yan Régis-Gianas. Quelle plateforme pour l'exercice en ligne ? Présentation disponible sur Didel, 2013.
- [2] Coursera, <https://www.coursera.org/>