

GÉNIE LOGICIEL 2 : SYNTHÈSE

Christian INGOUFF
Pierre-Alexandre TYNDAL
Sonia SEDDIKI
Yann CHARBONNIER

EISTI

2013/2014
Semestre 2
Jalon 5

Table des matières

1	Disclaimer	3
2	Introduction	4
3	Reformulation du cahier des charges	5
4	Analyse	8
4.0.1	Diagramme de cas d'utilisation	8
4.0.2	Diagramme de classes	8
4.0.3	Diagramme d'activité	11
5	Conception	12
6	Programmation	13
7	Travail en équipe	14
7.1	Présentation	14
7.1.1	Rôles	14
7.1.1.1	Dans la réalisation du projet	14
7.1.1.2	Dans l'équipe	15
7.2	Gestion du projet	16
7.2.1	Gestion du temps	16
7.2.2	Dynamique de l'équipe	17
7.3	Evolution de l'équipe	17
7.3.1	Rappel : cycle de vie de l'équipe	17
7.3.2	Evolution de l'équipe	17
7.4	Vision personnelle	17
7.4.1	Christian	17
7.4.2	Sonia	17
7.4.3	Pierre-Alexandre	17
7.4.4	Yann	18

Partie 1

Disclaimer

Grand disclaimer, ce rendu n'est pas contractuelle et n'est en aucun cas un rendu final de synthèse. La version finale arrive vers lundi, date instruite de rendu papier.

Partie 2

Introduction

Ce rapport présente une vue d'ensemble du projet et des réalisations en découlant. Nous reviendrons sur les différentes phases qui ont jalonné son déroulement, de la reformulation du cahier des charges jusqu'à la programmation du logiciel et son test.

Nous aurons également l'occasion, en dernière partie, de faire une rétrospective du travail en équipe mené tout au long du projet. La vision apportée sera plus axée sur la partie humaine ainsi que sur les rôles de chacun.

Partie 3

Reformulation du cahier des charges

La reformulation du cahier des charges est le point de départ de notre projet. Elle permet de s'assurer que chacun a bien compris les attentes vis à vis du logiciel demandé, et ainsi de poser des bases sûres pour les réalisations à venir.

Une étude rapide du projet en tant que futur logiciel informatique nous a permis d'isoler quelques contraintes implicites à respecter, dans une optique de rigueur, de facilité d'accès et d'optimisation :

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

Le cahier des charges impose aussi d'autres exigences :

- 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, créées ou supprimées doivent être mises à jour lors d'une sauvegarde.

Ci-dessous les principales fonctionnalités attendues. C'est la ligne directrice suivie pour l'élaboration du programme :

- 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

Pour perfectionner le programme, un peu austère et incomplet, nous avons prévu d'y ajouter certaines fonctionnalités supplémentaires. Elles sont facultatives car non exigées dans le cahier des charges, mais sont toutefois indispensables pour rendre le logiciel utilisables par tous et sans problème de sécurité :

- Authentification : Tout utilisateur devra s'authentifier afin de pouvoir utiliser le logiciel
- Interface graphique : Esthétique et utilisation aisée

Le cahier des charges une fois reformulé, les objectifs sont clairs. Nous pouvons maintenant procéder à l'analyse.

Partie 4

Analyse

La phase d'analyse va nous servir à effectuer une première modélisation du logiciel. Grâce à la création de multiples diagrammes, nous allons pouvoir aborder différentes parties de la création du logiciel, comme la création des éléments du projet ou encore le déroulement détaillé d'une fonction. Etant donné que nous allons programmer en Java, nous allons utiliser l'analyse UML (Unified Modeling Language) pour réaliser nos diagrammes.

4.0.1 Diagramme de cas d'utilisation

Ce diagramme définit les acteurs (Utilisateurs) ainsi que les différentes interactions possibles avec le logiciel (Cas d'utilisations). Il permet de voir clairement qui fait quoi (et par extension ce que tel ou tel utilisateur ne peut pas faire suivant son statut), mais également des points de passage obligés pour réaliser un cas d'utilisation. Par exemple, pour consulter les résultats d'une session QCM, il faut déjà avoir répondu à cette session de QCM. Enfin, nous y voyons les différentes extensions d'un cas d'utilisation : consulter une session QCM revient à visualiser le score et les réponses et/ou les statistiques.

4.0.2 Diagramme de classes

Nous posons ici les bases du code Java, qui est un langage orienté objet. Autrement dit, la programmation se fait en définissant :

- Des "classes" : les différentes entités de notre modèle, avec leurs attributs et méthodes associées
- Des "associations" entre ses classes : concrètement, ces liens se définissent par exemple par des appels d'instances de classe dans les fonc-

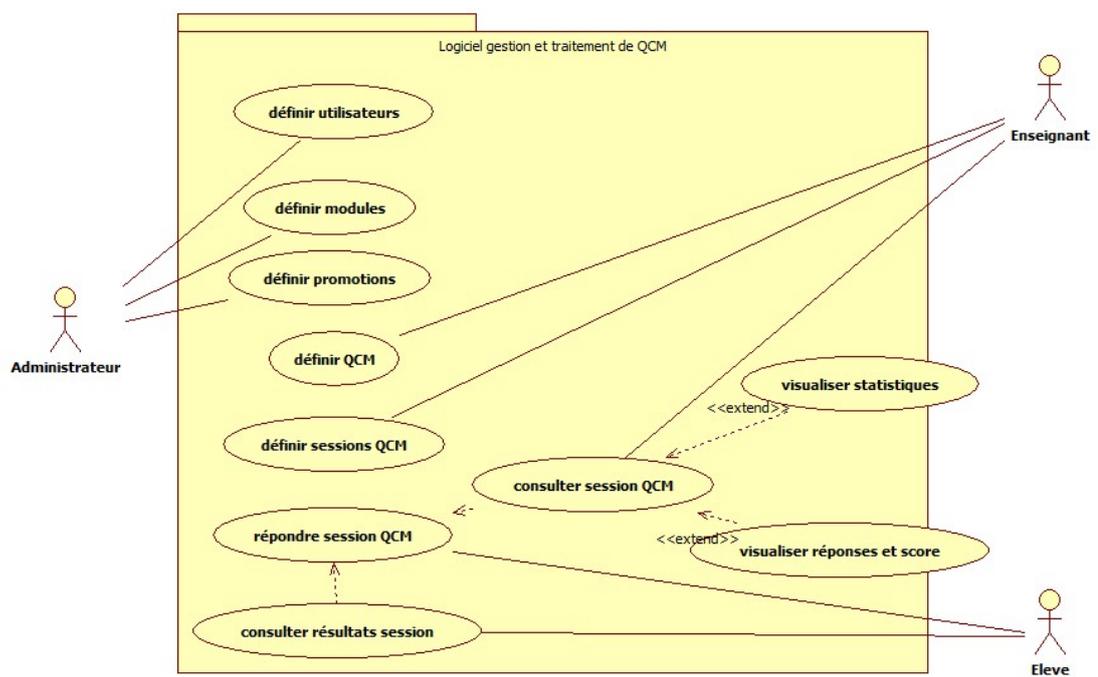


FIGURE 4.1 – Diagramme de cas d'utilisation

4.0.3 Diagramme d'activité

Partie 5

Conception

Partie 6

Programmation

Partie 7

Travail en équipe

7.1 Présentation

Cette partie se focalisera sur l'équipe en elle-même, ainsi que sur son fonctionnement tout au long du projet. Les principaux axes abordés seront les rôles tenus par chacun, la gestion du projet et enfin l'évolution de l'équipe au cours des mois.

7.1.1 Rôles

Comme énoncé dans le livre Team roles at work de Meredith BELBIN, il est important de distinguer le rôle d'une personne dans l'équipe de son rôle fonctionnel. Autrement dit, il faut faire la différence entre ce qu'elle fait d'un point de vue technique (son rôle fonctionnel dans le projet), et sa fonction dans l'équipe (son comportement au sein de l'équipe).

7.1.1.1 Dans la réalisation du projet

Dans l'ensemble, la répartition des tâches s'est faite en fonction des compétences et points forts de chacun afin de gagner en efficacité.

Voici un bref récapitulatif des acteurs du projet ainsi que leurs différentes compétences :

- INGOUFF Christian

Connaissances : Pascal, C, OCaml, HTML/CSS/PHP, SQL/gestion BDD, VBA

- TYNDAL Pierre-Alexandre
Connaissances : Pascal, C, Java, OCaml, HTML/CSS/PHP, SQL/gestion BDD
- SEDDIKI Sonia
Connaissances : C, Maple, SQL/gestion BDD
- CHARBONNIER Yann
Connaissances : C, SQL/gestion BDD

Etape 1 : Cahier des charges fonctionnel

La reformulation du cahier des charges a été faite par l'ensemble de l'équipe. Cela nous a permis de nous assurer que tout le monde comprenait bien les exigences ainsi que les différentes contraintes techniques imposées.

Etape 2 : Analyse

L'étape de l'analyse était conséquente, nous l'avons donc séparée en deux parties. Dans tous les cas, les tâches étaient réparties de sorte que chaque membre s'est attelé à la réalisation d'un type de diagramme en particulier. Toutefois, nous ne travaillions pas de manière isolée. En effet, l'avancée dans un type de diagramme pouvait révéler les lacunes d'un autre diagramme, et donc améliorer ce diagramme par la suite.

Etape 3 : Conception/Programmation

Cette étape a été retardée par rapport au planning initial, le diagramme de Gantt. Nous avons donc exploité aussi efficacement que possible les facilités de chacun pour rattraper ce retard.

Comme nous pouvons le constater, personne ne connaissait le Java avant le début du projet. Or c'était le langage de programmation imposé pour le logiciel. Toutefois, Christian et Pierre-Alexandre, en tant qu'ex-CPI, avaient une expérience de l'informatique et du code plus développée, ce qui leur permettait, pour une même tâche, de l'exécuter plus rapidement que des ex-CPGE. C'est pourquoi ils ont supervisé la partie programmation du projet. Yann, quant à lui, s'est occupé de toute l'interface graphique du logiciel (l'interface homme-machine), et Sonia de l'écriture des algorithmes des fonctions les plus complexes.

7.1.1.2 Dans l'équipe

Les rôles de chacun dans l'équipe sont un peu difficiles à déterminer, puisque non seulement ils n'étaient pas uniques, mais ils changeaient en fonc-

tion de l'étape traitée. Par exemple

Toutefois, nous pouvons distinguer un rôle principal basé sur le caractère pour chacun d'entre nous.

- Christian :
- Pierre-Alexandre : Souvent rêveur, apportait des idées
- Sonia :
- Yann :

7.2 Gestion du projet

Une autre partie de l'analyse de l'équipe est sa capacité à gérer l'avancement d'un projet et à le mener à bien. Ceci dépend à la fois de la gestion du temps (respect du planning, efficacité pendant les séances de travail) et de la dynamique de l'équipe (motivation, bonne entente, etc).

7.2.1 Gestion du temps

La gestion du temps disponible était prévue par le diagramme de Gantt en début de projet. Celui-ci tenait compte des différents événements qui pourraient ralentir l'avancée du projet (cours, vacances, campagnes BDE, etc), mais également les dates de soutenances et un planning prévisionnel de dates de rendu pour les différents livrables.

Le diagramme a été difficile à tenir, notamment par sa mauvaise conception. En effet, les notions qui allaient être abordées par la suite nous étaient totalement inconnues au moment où nous l'avons élaboré. Cela a conduit à une surestimation (ou sous estimation selon les cas) du temps nécessaire à la réamisation d'une tâche donnée. De plus, nous avons conçu notre planning de manière très linéaire et déconnectée : les différentes tâches se suivaient, et une fois l'une finie, on passait à la suivante. Les choses ont été bien différentes dans la réalité, puisqu'on n'a rarement pu considérer une phase comme totalement achevée. En effet, chaque nouvelle tâche révélait quelques oublis, lacunes, erreurs dans les tâches précédentes, d'autant plus que certaines d'entre elles étaient étroitement liées. Par exemple, pour l'analyse, certains types de diagrammes ne pouvaient pas être réalisés sans avoir fait un autre type de diagramme avant. Nous avons ainsi fait des allers-retours permanents entre les différentes parties, afin de les améliorer au maximum et de garder une

certaine cohérence.

7.2.2 Dynamique de l'équipe

7.3 Evolution de l'équipe

7.3.1 Rappel : cycle de vie de l'équipe

Une équipe possède différents stades de vie :

- Etape 1 : Constitution

- Etape 2 : Stabilisation

- Etape 3 : Intégration

- Etape 4 : Réalisation

- Etape 5 : Maturation

- Etape 6 : Dissolution

7.3.2 Evolution de l'équipe

7.4 Vision personnelle

7.4.1 Christian

7.4.2 Sonia

7.4.3 Pierre-Alexandre

Se basant essentiellement sur des notions en cours d'étude, ce projet a été compliqué à planifier. L'attente de connaissances a généralement joué en notre défaveur, nous mettant en retard en raison de sous estimation du temps de réalisation de certaines tâches. De plus, étant une poursuite partielle de l'étude de Génie Logiciel réalisée au premier semestre, la vision du projet était en partie déjà visualisée dans les esprits. Cependant, l'attente

de connaissances sur certains outils a fait en partie baisser le rythme et la motivation.

7.4.4 Yann

BLABLABLA

Partie 8

Conclusion