



École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique
4^e année
2014 - 2015

Cahier de Spécifications

Projet d'Ingénierie Logicielle

Enseignant

Nicolas RAGOT
nicolas.ragot@univ-tours.fr

Université François-Rabelais, Tours

Étudiants

Anaëlle HAMON
anaelle.hamon@etu.univ-tours.fr
Laura BELLEGO
laura.bellego@etu.univ-tours.fr
Jean COQUELET
jean.coquelet@etu.univ-tours.fr
Philippe LIM
philippe.lim@etu.univ-tours.fr
Florian GADAL
florian.gadal@etu.univ-tours.fr
Florian MONTALBANO
florian.montalbano@etu.univ-tours.fr
Ye TIAN
ye.tian@etu.univ-tours.fr

DI4 2014 - 2015

Version du 8 avril 2015

Table des matières

1	Introduction	5
2	Contexte de réalisation	6
2.1	Contexte	6
2.2	Objectifs	6
2.3	Bases méthodologiques	6
3	Description générale	7
3.1	Environnement du projet	7
3.2	Caractéristiques des utilisateurs	7
3.3	Contraintes de développement	7
3.3.1	Matérielles	7
3.3.2	Algorithmes	7
3.3.3	Environnements nécessaires	7
3.3.4	Délai de réalisation	8
3.4	Maintenance et évolution du système	8
4	Description des interfaces externes du système	9
4.1	Interfaces matériel/logiciel	9
4.2	Interfaces homme/machine	9
4.2.1	Interface enseignant	9
4.2.2	Interface administrateur	11
4.2.3	Interface logiciel/logiciel	11
5	Architecture générale du système	12
5.1	Éléments constituant la base de données	13
5.2	Éléments relatifs à la plateforme	13
6	Fonctionnalités	14
6.1	Authentification	14
6.2	Importation des résultats de tests	14
6.3	Suppression des données "trop vieilles"	14
6.4	Filtrage et requêtage	14
6.4.1	Filtrage personnalisé	15
6.4.2	Filtre sur le niveau CECRL avant/après	15
6.4.3	Filtrage sur le numéro étudiant	15
6.5	Génération de graphiques	15
6.5.1	Génération de camemberts	15
6.5.2	Génération de courbes	16
6.6	Envoi de l'historique aux étudiants	16
6.7	Rappel pour l'import	16
6.8	La maintenance de la base de données	16
6.9	Fonctionnalités supplémentaires	16



6.9.1	Génération de radars	17
6.9.2	Génération d'histogrammes	17
7	Conditions de fonctionnement	18
7.1	Performance	18
7.2	Capacité	18
7.3	Sécurité	18
8	Plan de développement	19
8.1	Sprint 1	19
8.2	Sprint 2	19
8.3	Sprint 3	19
8.4	Sprint 4	20
8.5	Diagramme de Gantt	20

Introduction

Ce document présente la conception d'un outil permettant de gérer les résultats de tests de langues, que des étudiants passent chaque semestre sur la plateforme Celene. Il devra permettre aux élèves de connaître leur niveau à la fin de chaque test et suivre leur niveau tout au long de leur cursus universitaire. De plus, il permettra aux enseignants d'accéder aux résultats des étudiants afin de réaliser des statistiques sur différents filtres personnalisés.

Ce projet a été proposé par Marjolaine Martin, coordinatrice des Centres de Ressources en Langues de l'université, Daniel Tron, responsable Centre de Ressources en Langues et est encadré par Nicolas Ragot, enseignant à Polytech Tours.

Contexte de réalisation

2.1 Contexte

Dès l'année prochaine, les étudiants de l'université de Tours passeront un test de niveau de langue sur Celene chaque semestre à partir de la L1 jusqu'au M2. Ce test permettra aux étudiants d'adapter leur apprentissage de la langue sur des points spécifiques, et de suivre leur progression entre chaque test. Ces tests existent déjà sur la plateforme web Celene, mais il manque encore la partie de suivi des résultats qui sera géré par notre outil.

2.2 Objectifs

L'objectif du projet est de créer une plateforme web, externe à Celene, qui sera alimentée par les résultats de chaque étudiant. Cette importation sera réalisée par une personne autorisée. Ces résultats seront sauvegardés dans une base de données. Ils seront ensuite récupérés afin de générer des statistiques pour chaque étudiant afin de lui envoyer un email, contenant un rapport récapitulant son niveau actuel et celui des années précédentes (sous forme de courbes).

Notre projet consiste donc à gérer le suivi d'un test en anglais (Cf. 3.4 Maintenance et évolution du système).

L'interface, réservée aux enseignants, permettra de générer des statistiques sur les niveau des étudiants sous forme de graphiques. Ces statistiques peuvent être configurées grâce à des filtres de base (UFR, groupe, compétence,...) et des filtres personnalisés pour faire une étude sur un groupe spécifique de personnes.

2.3 Bases méthodologiques

Dans le cas du développement d'une plateforme web, nous avons décidé d'utiliser les différents langages liés à ce type d'application (HTML, CSS, Javascript, PHP,...).

Pour ce qui est de la modélisation des fonctionnalités du système, nous avons choisi d'utiliser le langage UML et la méthode Merise pour la modélisation de la base de données.

Les sources du logiciel seront versionnées grâce à SVN géré depuis l'interface Redmine qui nous permet de gérer l'avancement du projet (Documentation, Rendez-vous, suivi de bugs,...).

Afin d'avoir un feedback régulier et de pouvoir fournir des livrables fréquemment, nous allons mettre en place une méthode agile pour le développement du projet.

Description générale

3.1 Environnement du projet

Notre outil se basera sur les tests effectués sur Celene, mais il sera complètement indépendant de cette plateforme. Seuls les exports des résultats seront utilisés pour communiquer entre les deux environnements.

Le site web et la base de données contenant entre autres les résultats seront hébergés sur un serveur. Théoriquement le serveur choisi doit être celui de l'université cependant, suite à des manques de sécurisation et avec la nécessité de multiples autorisations pour ce stockage, Polytech met à notre disposition un serveur virtualisé pour l'hébergement et le stockage des données de notre site.

3.2 Caractéristiques des utilisateurs

On peut différencier 4 types d'acteurs qui auront chacun des autorisations différentes sur la plateforme web :

- Les étudiants : Ils n'ont pas accès à la plateforme mais ils recevront un email lorsque leur résultat aura été traité et sauvegardé par l'outil.
- Les enseignants : Ils ont un accès restreint au système. Ils peuvent accéder aux résultats des étudiants de l'UFR auquel ils sont rattachés.
- "Accès supérieur" (responsables) : Ils pourront effectuer des requêtes sur l'ensemble des étudiants de l'université.
- Administrateur : L'administrateur pourra importer le fichier de résultats chaque semestre afin de lancer le processus de traitement et de l'envoi des emails aux étudiants et pourra également gérer les informations présentes sur la base de données : sauvegarde, modification, suppression, réinitialisation.

Pour l'authentification à la plateforme, un code commun sera fourni à tous les responsables, un code par UFR aux autres enseignants et un code pour l'administrateur. Au minimum une connexion par semestre de l'administrateur est nécessaire pour importer le fichier des résultats dans l'application.

3.3 Contraintes de développement

3.3.1 Matérielles

Nous avons besoin d'un accès à un serveur afin d'y stocker l'application et la Base de données, ainsi que d'un espace de stockage d'au minimum 5 Go. Le calcul de cet espace est donné dans les conditions de fonctionnement.

3.3.2 Algorithmes

Du point de vu des algorithmes, il faudra prendre en compte la complexité temporelle qui sera celle que l'on cherche à minimiser afin d'avoir une plateforme fluide du point de vue de l'utilisateur.

3.3.3 Environnements nécessaires

Nous avons la possibilité de réaliser soit une application lourde soit une plateforme web. Cependant l'application demande une installation poste par poste, ce qui est difficile à réaliser en vue du nombre

de postes concernés par l'application (personnels et/ou universitaires). De plus en vue de la nécessité du multi-plateforme, le développement d'une plateforme web a été préféré.

3.3.4 Délai de réalisation

Il s'agit d'un projet important aux yeux du CRL, ils souhaiteraient le mettre en place pour le premier semestre de l'année prochaine, ce qui implique pour nous que le projet soit terminé à la fin du semestre en cours.

3.4 Maintenance et évolution du système

A la fin du développement de cette application web, une documentation technique et utilisateur sera fourni, ce qui simplifiera une maintenance corrective et évolutive.

Pour cela, nous rédigerons des documents qui auront pour but de décrire l'ensemble des procédures d'installations à effectuer sur les postes des développeurs ainsi qu'un document concernant la configuration du serveur (dans le cas où le site web serait hébergé sur un nouveau serveur).

Nous prendrons en compte dans notre développement les différentes évolutions attendues par le client à long terme. En effet, dans le cadre d'une deuxième version de cette plateforme, notre client aimerait pouvoir générer de nouveaux types de graphiques.

De plus, les enseignants d'autres matières que l'anglais ont le souhait de créer leurs propres tests et d'en réaliser le suivi avec notre application. Ces autres matières peuvent concerner d'autres langues (allemand, chinois,...) ou d'autres matières générales (mathématiques, droit,...).

La gestion des droits d'utilisateur pour l'accès à l'application peut être amené à être affinée. Par exemple, ils seraient possible d'ajouter d'autres codes d'utilisateur afin de créer de nouveaux groupes avec des droits spécifiques sur l'application. Ceci pourra être fait depuis l'interface de l'administrateur.

Ces améliorations pourront être mise en place dans le cadre d'un projet visant à compléter cette application si le temps que nous avons pour le développement ne s'avère pas suffisant.

Description des interfaces externes du système

4.1 Interfaces matériel/logiciel

Dans ce projet, nous allons développer une interface web. Pour cela, l'utilisateur devra être muni d'un ordinateur et plus précisément d'un navigateur (avec JavaScript activé) et d'un service d'accès Internet afin de pouvoir accéder à la plateforme. De plus, le navigateur communiquera avec une base de données centralisée située sur un serveur virtualisé interne à l'Université. La plateforme sera donc accessible depuis l'intérieur du réseau universitaire.

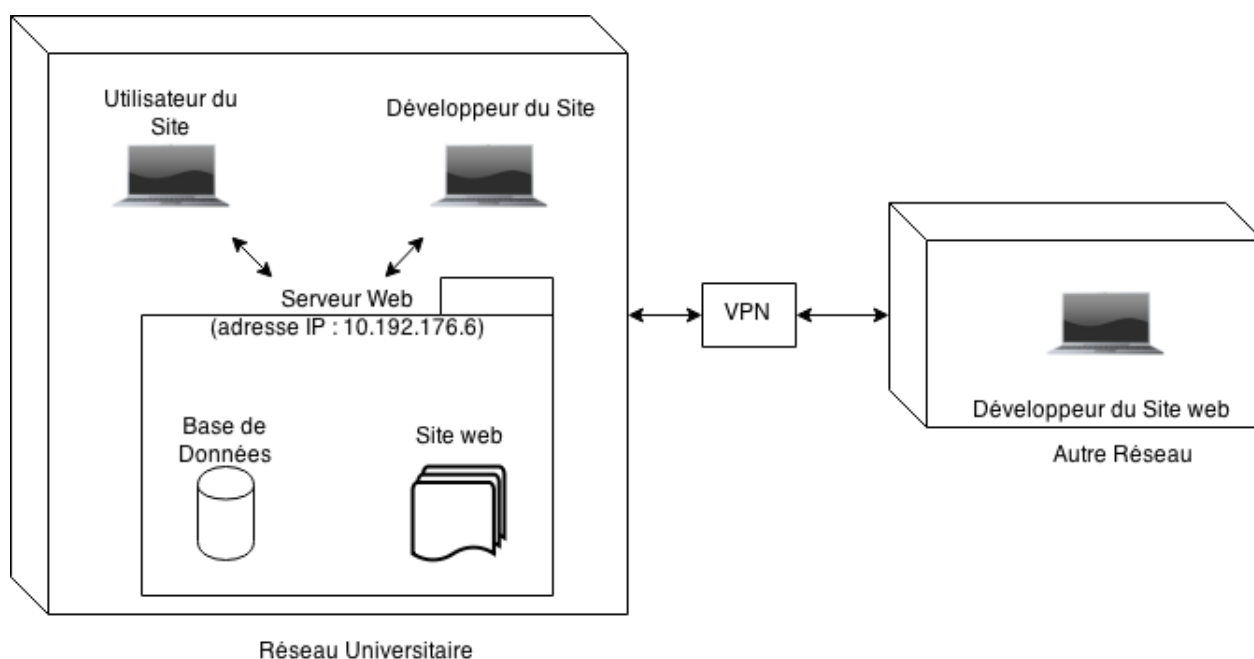


FIGURE 4.1 – Diagramme de Déploiement

4.2 Interfaces homme/machine

Les différents utilisateurs autorisés pourront avoir accès à la plateforme. Une gestion de l'authentification devra être mise en place pour différents types d'utilisateurs ayant des droits différents sur la plateforme.

4.2.1 Interface enseignant

Les enseignants auront accès, dans leur interface, à un ensemble de filtres et d'options leur permettant d'afficher ce qu'ils souhaitent. Les droits qui leur sont accordés leur restreindront, ou non, l'accès à certains filtres.

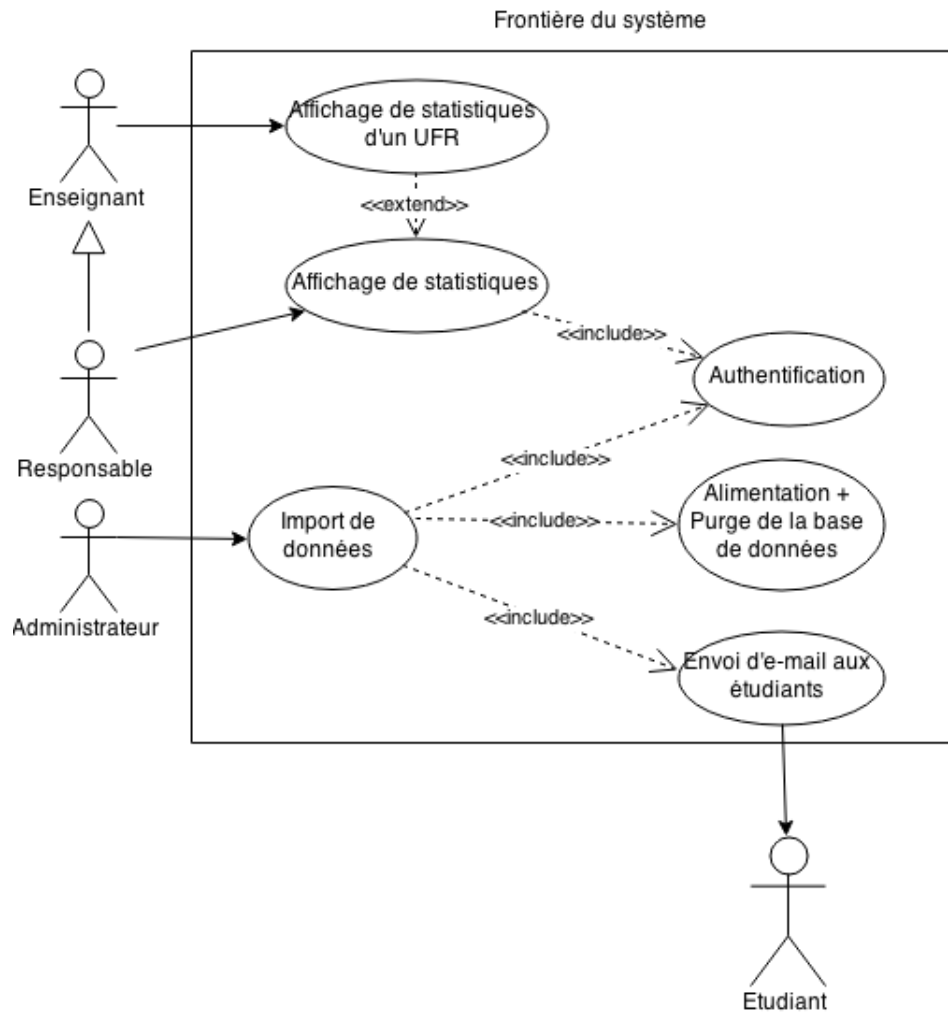


FIGURE 4.2 – Diagramme de contexte

Filtres

Informations générales

UFR
Art et Sciences Humaines

Département
Psychologie

Filtres supplémentaires

Filtre personnalisé

Statuts spéciaux
SSE Etraum

Filtres sur les niveaux d'études

Année d'étude
L1 L2 L3 MI MD

Choix de la période
Année ponctuelle
Plage d'années De à

Affichages en Camemberts

Année
L1 L2 L3 MI MD

Semestre
S1 S2

Filtre sur le niveau : An départ (premier test) A l'arrivée (dernier test)

Affichages en Courbes

Debut

Fin

Année
L1 L2 L3 MI MD

Semestre
S1 S2

Compétences à afficher : Grammaire Syntaxe Vocabulaire Compréhension Orale Moyenne Orale

Filtre sur le niveau : An départ (premier test) A l'arrivée (dernier test)

Afficher

FIGURE 4.3 – Maquette de l'interface enseignant

4.2.2 Interface administrateur

L'interface de l'administrateur permettra d'avoir accès à l'ensemble des fonctionnalités liées à l'administration de la plateforme : import de résultats, ajout de données, sauvegarde de la base de données etc.



Importation et gestion de la base de données

Importer de nouveaux resultats

Fichier de résultats : Aucun fichier choisi

Fiches Celene : Aucun fichier choisi

Matière du test :

Année :

Semestre : S1 ☐ S2 ☐

Gestion de la base de données

Sauvegarde/réinitialisation des données

Lieu de sauvegarde :

Fichier de la base de données : Aucun fichier choisi

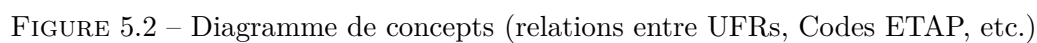
Modification des données

FIGURE 4.4 – Maquette de l'interface administrateur

4.2.3 Interface logiciel/logiciel

Concernant les interfaces logiciel/logiciel, il s'agit d'abord de la liaison entre la base de données et notre application (situées sur le même serveur). En effet, étant donné que nous manipulons une base de donnée il faudra régulièrement communiquer avec celle-ci.

De plus, nous utiliserons des fichiers exportés depuis Celene on peut donc également considérer ces fichiers comme interface entre Celene et notre base de données. Le format de ces fichiers devra donc être précisé par la suite.



12

5.1 Eléments constituant la base de données

Parmi ces éléments, on a, entre autres, les informations sur les tests :

- les informations des pré-tests : numéro étudiant, date de l'import des données (clé utilisée pour lier les pré-tests aux résultats bruts des tests), filtre personnalisé, langue du test.
- les résultats au test : numéro Etudiant, date de l'import des données (utilisé pour purger la BD), code ETAP (cohorte = UFR+Département+Niveau), note totale (en pourcentage), note à chaque compétence.

Lors de l'alimentation les informations des tests et des pré-tests seront joints dans la même table grâce au numéro étudiant unique. Notre base de données contiendra également des tables satellites :

- Code ETAP : correspondance entre le code ETAP et la cohorte
- La liste des UFR
- La liste des composantes (liées aux UFR)
- Les codes d'accès liés aux UFR
- La répartition entre les niveaux CECRL et les résultats aux tests par compétence. Pour les tests d'anglais, les compétences sont le vocabulaire, la grammaire, la syntaxe et la compréhension orale.

5.2 Eléments relatifs à la plateforme

Il s'agit des éléments que l'utilisateur manipulera directement depuis la plateforme web et qui interagissent avec des éléments de la base de données.

- Les filtres : sélectionnés par l'utilisateur, ils permettront de construire les requêtes
- Les requêtes : elles regroupent l'ensemble des filtres choisis par l'utilisateur et sont envoyées à la base de données pour un affichage des résultats.
- Les graphiques : il s'agit d'images générées à partir des résultats des requêtes et qui représenteront ces résultats sous forme de diagrammes ou de courbes.
- L'historique de chaque étudiant : constitué d'un graphe et d'un commentaire neutre (sans jugement), il s'agit de ce qui sera envoyé aux étudiants passant les tests.

Fonctionnalités

6.1 Authentification

L'authentification va permettre aux utilisateurs de se connecter. Celle-ci se fera via un code d'accès unique pour tous les enseignants d'un même UFR. Il existera un code pour l'ensemble des personnes "responsables" (responsables CRL) et un autre pour l'administrateur de la plateforme.

L'utilisateur devra, à chaque fois qu'il tente d'accéder à la plateforme, rentrer le code d'accès en sa possession. Si le code fourni est correct, l'accès à l'interface sera autorisé dans le cas contraire l'accès lui sera refusé. Pour pouvoir savoir si l'utilisateur a bien accès à l'interface cela se fera via une requête à la base de données.

6.2 Importation des résultats de tests

Cette fonctionnalité concerne l'importation des nouveaux résultats de tests passé sur Celene dans la base de données.

Chaque fin semestre l'administrateur va importer les nouvelles données grâce à deux fichiers, le premier contenant les données des étudiants (fichier avec l'extension ods), le deuxième fichier sera le fichier de prétest (fichier avec l'extension ods). Il devra également renseigner plusieurs champs :

- L'année scolaire (ex : 2014-2015)
- Le semestre concerné (S1 ou S2)
- La matière du test (dans un premier temps, juste l'anglais)

Tant que l'e-mail contenant l'historique de chaque étudiant ne sera pas envoyé, il sera possible d'importer à nouveau des fichiers pour une matière, une année et un semestre. Cet import sera cependant bloqué à partir du moment où les mails qui concernent ce test auront été envoyés.

Avant l'import des données, l'administrateur pourra choisir un emplacement sur son ordinateur pour sauvegarder une version de la base de données. Ainsi, en cas de problèmes décelés lors de l'importation, l'administrateur pourra charger la base sauvegardée auparavant. Il s'agit d'une sauvegarde.

6.3 Suppression des données "trop vieilles"

Cette fonctionnalité va nous permettre d'effacer les données "trop vieilles" de notre base de données. Il s'agit des données qui concernent des tests passé il y a plus de 10 ans.

A chaque importation de nouvelles données, il faudra vérifier qu'il n'y a pas des données dépassées dans la base de données. S'il y en a nous les supprimerons.

Ceci est nécessaire pour que notre base de données ne deviennent pas trop chargée au fil du temps.

6.4 Filtrage et requêtage

Cette fonctionnalité va permettre aux utilisateurs de filtrer les étudiants par UFR, département, année d'étude, année concernée et statuts spéciaux, et ainsi d'utiliser ces filtres pour construire nos requêtes à la base de données.

Le responsable du CRL aura tous les droits de filtre (UFR, département, statuts spéciaux, niveau de départ ou d'arrivée ou le filtrage sur un étudiant en particulier) alors que l'enseignant pourra quand à lui filtrer uniquement les résultats liés à son UFR..

Les utilisateurs pourront choisir plusieurs années d'études (L1 et L3) sur une année scolaire (ex : 2014) mais aussi sur une période (ex : de 2011 à 2015).

6.4.1 Filtrage personnalisé

Cette fonctionnalité va permettre aux utilisateurs de définir eux même un ou plusieurs filtres. Le nombre maximum de filtres personnalisés que l'on pourra ajouter est de 5 . Pour cela, Les enseignants devront faire remplir un champs aux étudiants dans le pré-test. Le filtrage se fera donc sur ce champs.

La chaine de caractères entrée devra suivre une nomenclature fixée au préalable afin de ne pas avoir de doublons.

Exemple

2 enseignants veulent tester leurs élèves étrangers et font remplir le champs qui concerne le filtre personnalisé à leurs étudiants avec la chaine de caractères : "ETRANGER". Lors du filtrage, aucune distinction ne sera faite entre les étudiants des deux enseignants. C'est pour cela qu'il faudra respecter une nomenclature définie pour l'ensemble des enseignants.

6.4.2 Filtre sur le niveau CECRL avant/après

Il s'agit ici d'introduire un nouveau filtre afin de préciser, lors de notre recherche, que l'on s'intéresse aux étudiants ayant un niveau au départ (niveau évalué au premier test) ou un niveau à l'arrivée (niveau au dernier test) précis.

Exemple

On pourra ainsi voir la progression des étudiants qui ont eu un niveau A2 au premier test.

6.4.3 Filtrage sur le numéro étudiant

Ce filtre permet d'afficher les statistiques d'un étudiant. Il remplace l'ensemble des filtres expliqué ci-dessus. Cet unique filtre nous permettra de suivre un étudiant. Les graphiques adaptés sont les courbes pour une progression ou les radars (non prioritaire dans le développement) pour avoir le niveau de l'étudiant dans chacune des compétences à un instant donné.

6.5 Génération de graphiques

Cette fonctionnalité en regroupe deux : la génération de camemberts et la génération de courbes. Le but de cette fonctionnalité générale est d'avoir un format de nos images facilement exportable : image.

On prendra en compte les compétences pour l'affichage de résultats. On pourra aussi choisir de prendre en compte une moyenne pondérée des compétences. La pondération prise en compte pourra être modifiée dans l'interface administrateur et sera initialisée à 1 pour chaque compétence.

6.5.1 Génération de camemberts

Cette fonctionnalité va nous permettre d'afficher la proportion des niveaux CECRL(A2, A1, B2, B1, C2) à un instant donné pour l'ensemble des étudiants répondant aux critères présents dans les autres filtres. Il s'agit ici d'afficher un camembert pour une compétence choisie. Le nombre d'étudiants concerné sera indiqué.



6.5.2 Génération de courbes

Il s'agit d'un affichage de l'évolution des niveaux pour chaque compétence que l'utilisateur aura sélectionnée.

L'utilisateur pourra choisir s'il souhaite afficher le niveau CECRL ou le pourcentage obtenu pour chaque compétence.

Les courbes permettent d'afficher une progression, en conséquence l'utilisateur devra indiquer la période à afficher ainsi que les compétences à représenter sur le graphe. Ensuite l'utilisateur pourra visualiser les courbes par compétence et le nombre d'étudiants concernés par le résultat à un instant précis.

6.6 Envoi de l'historique aux étudiants

On souhaite ici générer un document qui contient le résultat de l'étudiant et lui envoyer par mail. Le résultat de l'étudiant sera présenté sous forme de courbes de compétences. Les points de la courbe seront des cercles qui contiendront le résultat à chaque semestre.

Les mails pourront être envoyés une fois que les données auront été importées. Cet envoi pourra être fait soit directement après l'import soit plus tard. Cependant, les mails ne pourront être envoyés qu'une seule fois et une fois envoyés, les résultats concernés ne pourront plus être modifiés dans la base de données.

6.7 Rappel pour l'import

Le but ici est de générer un mail qui sera envoyé à l'administrateur à la fin de chaque semestre pour lui rappeler d'importer les résultats des tests. L'adresse mail de l'administrateur devra donc être sauvegardée dans notre base de données.

Une adresse supplémentaire sera sauvegardée afin que 2 personnes soient averties (pour plus de sécurité).

La date de l'envoi de l'e-mail de rappel sera fixée pour le semestre 1 et le semestre 2 mais pourra être modifiée par l'administrateur.

6.8 La maintenance de la base de données

En cas de problème, l'utilisateur pourra ré-importer les données qu'il aura sauvegardées avant l'import de nouveaux résultats. Il pourra également avoir accès aux logs qui concernent les dernières manipulations de la base de données : imports de résultats, envois d'e-mails, modifications de données.

Cette fonctionnalité ne sera accessible qu'à partir de l'interface administrateur.

Remarque

L'utilisateur visualisera un avertissement lorsqu'il voudra réimporter les données dans la base de données afin d'être sûr qu'il ait bien sauvegardé les données au préalable.

6.9 Fonctionnalités supplémentaires

Les fonctionnalités supplémentaires concernent des nouveaux affichages possibles qui pourront être ajoutés à la plateforme une fois que les fonctionnalités présentées ci-dessus auront été implémentées et validées.

6.9.1 Génération de radars

Il s'agit de l'affichage d'un graphique radar qui représente les niveaux CECRL moyens par compétence.

L'utilisateur visualisera un graphique radar en fonction des niveaux moyens par compétence du groupe d'étudiant, qui sera préalablement créée grâce aux filtres.

6.9.2 Génération d'histogrammes

Il s'agit de l'affichage d'un histogramme en fonction des niveaux(A1,C1,etc...), le nombre d'étudiants concernés par le niveau et l'effectif total.

Conditions de fonctionnement

7.1 Performance

Concernant les performances, rien n'est précisé par le client concernant l'import des données ou l'envoi des e-mails aux étudiants. Cependant, un temps de requêtage et d'affichage des graphiques court est attendu : moins de 5 secondes pour l'affichage d'un graphiques.

7.2 Capacité

Une entrée dans la base est composée de notes (float), de feedbacks et de chaînes de caractères correspondant aux différentes informations sur l'étudiant. Ce qui fait 5 kilooctets par entrée. Nous arrondissons le nombre d'étudiants à 50 000. Avec la sauvegarde de données sur 10 ans, pour un type de test (d'anglais), on aurait besoin au minimum de 2.56 Go.

Nous avons donc pris en compte un espace de 5Go au total par langue afin de ne pas avoir de problème d'espace de stockage.

7.3 Sécurité

Etant donné que la plateforme sera déployée dans un premier temps à l'intérieur du réseau universitaire, la plateforme ne sera pas sensible aux failles de sécurité. Ce point implique donc que les enseignants ne pourront avoir accès à la plateforme que depuis un poste de l'université.

Plan de développement

Etant donné que nous avons choisi de travailler avec une méthode agile, cela signifie que le client aura des retours réguliers de ce que l'on aura produit.

Chaque retour correspondra à un ensemble de tâches développées et intégrées à notre plateforme qui constituera notre livrable ainsi que des documents explicitant le travail effectué et les différentes configurations ou étapes nécessaires pour avoir accès aux fonctionnalités développées. Pour chacune de ces phases, une documentation technique et utilisateur sera fournie.

Dans les parties qui suivent, nous allons expliciter les différents sprints de notre projet avant de les représenter dans un diagramme de Gantt pour les répartir dans le temps.

8.1 Sprint 1

La première tâche (premier sprint) consiste à mettre en place notre plateforme web ainsi qu'à développer les premières fonctionnalités de notre projet. Durant cette tâche, nous allons mettre en place la base de données ainsi que l'interface web. Nous allons aussi configurer notre serveur pour l'hébergement du site web. Enfin, nous allons développer les fonctionnalités d'authentification ainsi que d'import de résultats dans la base de données.

A la fin de ce sprint, une interface validée et fonctionnelle sera en place faisant la liaison avec la base de données. L'authentification fonctionnera correctement afin d'assurer l'accès sécurisé à la plateforme et l'import des données pourra être fait depuis l'interface de l'administrateur. Plusieurs documents seront fournis : la liste des codes d'authentifications par UFR, l'adresse d'accès au site web ainsi que l'ensemble des informations nécessaires pour une maintenance correcte de la plateforme.

8.2 Sprint 2

Durant cette phase, nous allons développer les fonctionnalités principales de l'interface d'un enseignant : la gestion des filtres et l'affichage des graphiques. Il s'agit de la tâche la plus importante car elle regroupe l'ensemble des fonctionnalités qui seront utilisées par tous les utilisateurs de la plateforme. Il faudra également effectuer une validation du formulaire afin que tous les champs nécessaires soient remplis.

A la fin de ce sprint, l'interface sera donc munie des filtres et affichages principaux parmi ceux énoncés dans ce cahier des spécifications soient tous les filtres hormis les filtres personnalisés et les affichages développés durant cette phase seront les courbes et les camemberts.

8.3 Sprint 3

Ce sprint consiste à développer l'ensemble des fonctionnalités relatives à l'interface de l'enseignant. Ces fonctionnalités regroupent donc la gestion de la base de données (sauvegarde, chargement, modification) ainsi que l'envoi des rapports aux étudiants.

Après le développement de ce sprint, les fonctionnalités liées à l'interface administrateur seront développées. Le livrable contiendra également la procédure que l'administrateur devra suivre. L'interface sera donc, après ce sprint, totalement opérationnelle. Il ne manquera que le développement des fonctionnalités supplémentaires.

8.4 Sprint 4

Ce dernier sprint regroupe l'ensemble des fonctionnalités à priorité plus faibles qui n'ont pas été développées durant les autres phases du projet : filtres personnalisés, affichage en radar, histogrammes.

L'interface disponible après cette phase sera donc complète.

8.5 Diagramme de Gantt

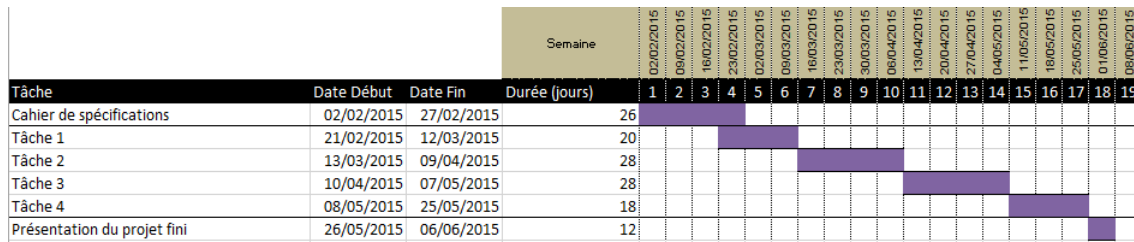


FIGURE 8.1 – Diagramme de Gantt

Si on considère qu'un jour/homme est égal à 4h/jour, la durée indiquée sur ce diagramme correspond au temps prévu au total (cumulé avec tous les développeurs) pour chaque sprint. Cependant, nous avons intégré pour la période du sprint 1 une phase d'installation et de formation donc le temps passé sur ce sprint peut être plus élevé.

Projet d'Ingénierie Logicielle

Département Informatique
4^e année
2014 - 2015

Cahier de Spécifications

Résumé : Dans le cadre du Projet d'Ingénierie Logicielle (PIL), nous travaillons sur le développement d'une plateforme web visant à afficher des résultats de tests de langues. Ce document a pour but de définir le plus précisément possible les attentes de nos clients.

Enseignant

Nicolas RAGOT
nicolas.ragot@univ-tours.fr

Université François-Rabelais, Tours

Étudiants

Anaëlle HAMON
anaelle.hamon@etu.univ-tours.fr
Laura BELLEGO
laura.bellego@etu.univ-tours.fr
Jean COQUELET
jean.coquelet@etu.univ-tours.fr
Philippe LIM
philippe.lim@etu.univ-tours.fr
Florian GADAL
florian.gadal@etu.univ-tours.fr
Florian MONTALBANO
florian.montalbano@etu.univ-tours.fr
Ye TIAN
ye.tian@etu.univ-tours.fr

DI4 2014 - 2015