

Rapport de Conception

PAR T.RAMONI, M.ZOUHIRI, U.COUDRAY & N.AZZA.

08/12/2014

Résumé :

Ce document est un rapport de conception d'un système de gestion de donnée d'un institut tels que l'IUT de Vélizy. Il présentera une partie de conception générale dans lequel le sujet sera explicité par un diagramme de classe et de package. Et par la réalisation des cas d'utilisation. De plus, on présentera nos méthodes de développement. Ensuite, une partie conception détaillée où on s'intéressera au comportement du système en profondeur. Enfin, une partie dans laquelle on effectuera les différents tests pour s'assurer du bon fonctionnement de notre application.

Table des matières

1.Introduction.....	4
1.1.Objectif de ce document.....	4
1.2.Présentation du projet.....	4
1.3.Sujet.....	4
1.4.Description du rapport.....	4
2.Conception Générale.....	6
2.1.Lecture de l'énoncé.....	6
2.2.Cas d'utilisation niveau stratégique.....	6
a)Cas d'utilisation 1 : Gérer la base de donnée de l'IUT.....	6
b)Cas d'utilisation 2 : Gérer la base de donnée des Étudiants.....	6
c)Cas d'utilisation 3 : Gérer la base de donnée des Département.....	6
d)Cas d'utilisation 4 : Gérer la base de donnée des Enseignants.....	6
e)Cas d'utilisation 5 : Gérer la base de donnée des Cours.....	6
2.3.Cas d'utilisation niveau utilisateur.....	7
a)Cas d'utilisation 6 : Gérer un institut.....	7
b)Cas d'utilisation 7: Gérer un étudiant.....	7
c)Cas d'utilisation 8: Gérer un département.....	7
d)Cas d'utilisation 9: Gérer un enseignant	7
e)Cas d'utilisation 10: Gérer un cours.....	7
2.4.Diagramme de classe.....	8
2.5.Autres diagrammes.....	9
2.6.Organisation des développements.....	10
2.7.Description des outils et de la plate-forme de développement.....	10
3.Conception détaillée.....	11
3.1.Diagramme de classe.....	11
3.2.Diagramme d'interaction.....	11
3.3.Diagramme d'objet.....	11
4.Fiche de Tests :	12
4.1.Tests Unitaires :	12
a)Résultats des tests unitaires.....	12
Classe : Institut 06/12/2014.....	12
Classe : Departement 06/12/2014.....	13
Classe : Cours 06/12/2014.....	13
Classe : Enseignant 06/12/2014.....	14
Classe : Etudiant 06/12/2014.....	14
4.2.Test de validation.....	15
4.3.Rapport d'erreur.....	15
5.Conclusion.....	16

Introduction

1. Introduction

1.1. Objectif de ce document

...

1.2. Présentation du projet

Le projet à réalisé est une application de la gestion d'un institut tel que l'IUT de Vélizy. Cette application permettra de gérer les différentes composantes de l'institut : les étudiants, les départements, les professeurs et les cours.

1.3. Sujet

Un IUT possède des départements. Chaque département a des professeurs a disposition pour enseigner les cours qu'il doit dispenser. Les étudiants faisant partie de l'IUT suivent les cours.

L'IUT a un nom, une adresse et un numéro de téléphone. Il peut ajouter un étudiant ou perdre un étudiant. Il peut accéder aux informations d'un ou de tous ses étudiants. Il en est de même avec ses départements.

Chaque étudiant possède un nom et un numéro d'étudiant.

Chaque département peut ajouter des enseignants ou perdre des enseignants. Il peut également accéder aux données d'un ou de tous les enseignants. Les enseignants ont un nom.

Un des enseignants est désigné comme étant le directeur du département.

1.4. Description du rapport

Tout d'abord, la conception générale où nous présenterons les cas d'utilisation niveau stratégique et utilisateur. Puis nous effectuerons une approche orientée objet pour la modélisation du système, ensuite nous parlerons de l'organisation des développements et enfin nous décrirons les outils ainsi que la plate-forme de développement.

Dans une troisième partie, nous verrons la conception détaillée qui est composé du diagramme de classe (à un niveau d'abstraction proche du code), et nous décrirons le comportement du système

avec un diagramme d'interaction niveau bas.

Dans une quatrième partie nous effectuerons les différents types de tests : Tests Unitaires, Tests de Validation, Rapport d'erreur.

Enfin nous donnerons dans une cinquième partie une synthèse de ce qui a été fait.

On commencera par présenter la phase de conception du projet avec : les classes, les objets, les exigences, les diagrammes de classes, les scenarii et leur diagrammes de séquences.

On testera ensuite notre application à l'aide des cas de tests.

Conception Générale

La conception générale va nous permettre de comprendre la méthode de conception de notre application, le raisonnement suivi. Nous pourrons grâce à ce document trouver tous les renseignements permettant ultérieurement de modifier, corriger, faire évoluer et réutiliser le travail déjà effectué. Ce document permettra également d'avoir une première approche des objets qui vont être instanciés, leurs relations, leurs rôles, et fera ainsi gagner du temps pour la phase de programmation.

2. Conception Générale

2.1. Lecture de l'énoncé

.....

2.2. Cas d'utilisation niveau stratégique

a) Cas d'utilisation 1 : Gérer la base de donnée de l'IUT

Niveau: Stratégique

b) Cas d'utilisation 2 : Gérer la base de donnée des Étudiants

Niveau: Stratégique

c) Cas d'utilisation 3 : Gérer la base de donnée des Département

Niveau: Stratégique

d) Cas d'utilisation 4 : Gérer la base de donnée des Enseignants

Niveau: Stratégique

e) Cas d'utilisation 5 : Gérer la base de donnée des Cours

Niveau: Stratégique

2.3. Cas d'utilisation niveau utilisateur

a) Cas d'utilisation 6 : Gérer un institut

Niveau: Utilisateur

Scénario nominal: 1 – Le gérant ajoute un institut
2 – Le gérant ajoute des départements à l'institut
3 – Le gérant ajout des étudiants à l'institut

b) Cas d'utilisation 7: Gérer un étudiant

Niveau: Utilisateur

Scénario nominal: 1 – Le gérant ajoute un étudiant
2 – Le gérant effectue les modifications nécessaires
3 – Le gérant supprime un étudiant

c) Cas d'utilisation 8: Gérer un département

Niveau: Utilisateur

Scénario nominal: 1 – Le gérant ajoute un département
2 – Le gérant effectue les modifications nécessaires
3 – Le gérant supprime un département

d) Cas d'utilisation 9: Gérer un enseignant

Niveau: Utilisateur

Scénario nominal: 1 – Le gérant ajoute un enseignant
2 – Le gérant effectue les modifications nécessaires
3 – Le gérant supprime un enseignant

e) Cas d'utilisation 10: Gérer un cours

Niveau: Utilisateur

Scénario nominal: 1 – Le gérant ajoute un cours
2 – Le gérant effectue les modifications nécessaires
3 – Le gérant supprime un cours

2.4. Diagramme de classe

Ce diagramme de classe présenté la modélisation du système à un niveau d'abstraction élevé donc proche de l'utilisateur.

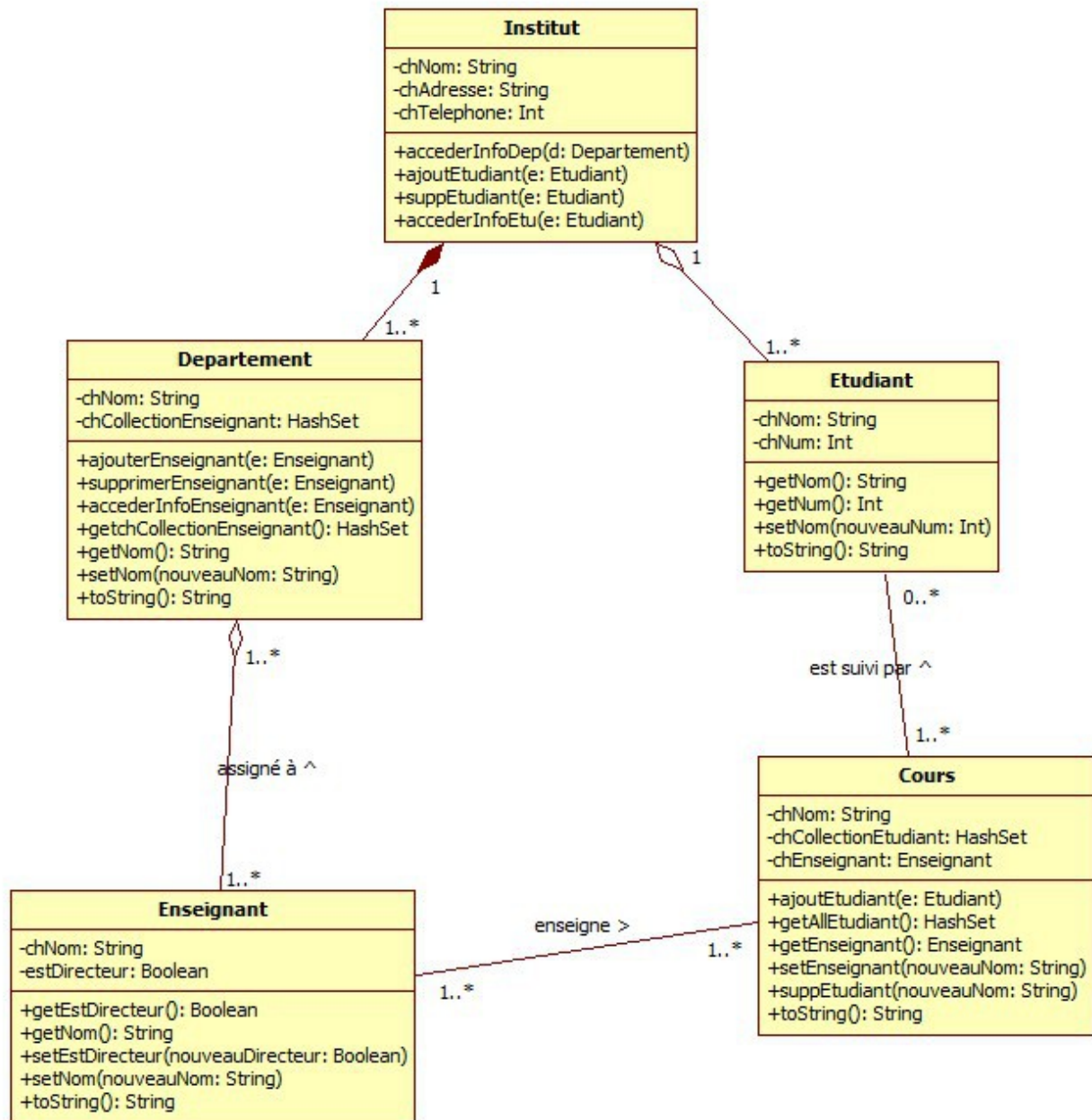


Figure 1 – Diagramme UML (bas niveau) du projet IUT

2.5. Autres diagrammes

Diagramme de package + diagramme de cas d'utilisation ??

2.6. Organisation des développements

Le processus unifié (ou *unified process* en anglais) est la méthode de conception logicielle que nous avons utilisé dans le cas de la programmation de l'application de gestion de l'institut.

2.7. Description des outils et de la plate-forme de développement

Lors de la phase de conception, de programmation et de tests, nous avons utilisé différents outils afin de pouvoir travailler :

- starUML : un logiciel qui permet de réaliser simplement des diagrammes UML tels que les diagrammes de classe ou d'objet.

- Eclipse : un logiciel qui nous a permis de compiler, de tester et d'exécuter nos programmes.

- GitHub : une application de gestion de version qui facilite la programmation.

Les langages utilisés sont les suivants :

- Java est un langage orientée objet et qui est parfaitement adaptée au cas de système de gestion d'un institut.

- JUnit intégré dans Eclipse nous a permis de réaliser les tests unitaires.

- JavaDoc qui sert à commenter le code et ainsi le rendre compréhensible pour d'autre développeur.

Conception détaillée

3. Conception détaillée

3.1. *Diagramme de classe*

Voici ci-dessous le diagramme de classe à un niveau d'abstraction bas (proche du code):

→ Reverse engineering

3.2. *Diagramme d'interaction*

Voici les diagrammes d'interaction présentant les différents cas d'utilisation de l'application :

3.3. *Diagramme d'objet*

Fiche de Tests

Voici les fiches de test permettant de regrouper les informations d'un scénario avec l'action de l'utilisateur et le résultat attendue, elles nous serviront plus tard pour tester notre application lorsque celle-ci sera fini.

4. Fiche de Tests :

4.1. Tests Unitaires :

Des tests unitaires se sont déroulés pendant toute la phase de codage.
Ce document représente donc que la compilation des tests unitaires de chaque classe.

Toutes les classes testés ont été testés à leur dernière version (dernière date de modification).
Ces tests unitaires formalisés se sont déroulés du 01/11/2014 au 07/12/2014 sur une machine hôte à l'aide de JUNIT sur Eclipse 4.4 Luna.

La mise en forme complète de ce document a été réalisée en même temps que les tests sur la même machine (mise à jour du texte - correction d'anomalies - ...).

Pour connaître la liste des tests unitaire, il suffit de se reporter au sommaire de ce document.

Les tableaux d'essais sont calqués sur le modèle suivant :

<u>Classe</u> :	NOM DE LA CLASSE	DATE DE LA DERNIERE MODIFICATION
<u>Rôle</u> :	Rôle de la classe testé (en quelques lignes)	
<u>Entrées</u> :	Entrée(s) de la classe testée	
<u>Sorties</u> :	Sortie(s) de la classe testée	
CONTROLES :		RESULTATS :
Définition des actions à vérifier.		OK - NOK

Tableau 1 – Modèle des Tests

a) Résultats des tests unitaires

<u>Classe</u> : <i>Institut</i> 06/12/2014	
<u>Rôle</u> : Classe représentant un institut qui contient des départements et des étudiants.	
<u>Entrées</u> : -	
<u>Sorties</u> : -	
CONTROLES :	RESULTATS :
Ajout Departement à l'institut	OK
Ajout Etudiant à l'institut	OK
Supprimer Departement à l'institut	OK
Supprimer Etudiant à l'institut	OK
Modifier Nom de l'institut	OK
Modifier Adresse de l'institut	OK
Modifier Numéro de téléphone de l'institut	OK

Tableau 2 – Tests Unitaires de la classe Institut

<u>Classe</u> : <i>Departement</i> 06/12/2014	
<u>Rôle</u> : Classe représentant un département qui contient des enseignants.	
<u>Entrées</u> : -	
<u>Sorties</u> : -	
CONTROLES :	RESULTATS :
Ajout Enseignant au Departement	OK
Supprimer Enseignant au Departement	OK
Modifier Nom du Departement	OK

Tableau 3 – Tests Unitaires de la classe Departement

<u>Classe</u> : <i>Cours</i> 06/12/2014	
<u>Rôle</u> : Classe représentant un cours qui contient des élèves et un professeur .	
<u>Entrées</u> : -	
<u>Sorties</u> : -	

CONTROLES :	RESULTATS :
Modifier Enseignant au Cours	OK
Ajouter Etudiant au Cours	OK
Supprimer Etudiant au Cours	OK
Modifier Nom du Cours	OK

Tableau 4 – Tests Unitaires de la classe Cours

<u>Classe :</u> Enseignant 06/12/2014	
<u>Rôle :</u> Classe représentant un enseignant.	
<u>Entrées :</u> -	
<u>Sorties :</u> -	
CONTROLES :	RESULTATS :
Modifier Nom de l'Enseignant	OK
Désigner comme directeur	OK

Tableau 5 – Tests Unitaires de la classe Enseignant

<u>Classe :</u> Etudiant 06/12/2014	
<u>Rôle :</u> Classe représentant un etudiant.	
<u>Entrées :</u> -	
<u>Sorties :</u> -	
CONTROLES :	RESULTATS :
Modifier Nom de l'Etudiant	OK
Modifier Numero de téléphone de l'Etudiant	OK

Tableau 6 – Tests Unitaires de la classe Etudiant

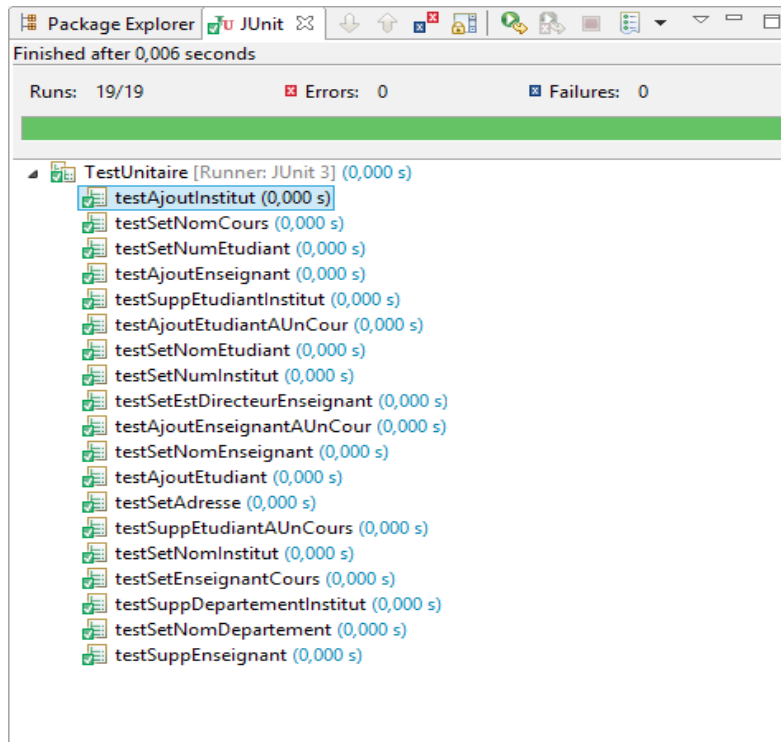


Image 1 – Validation des tests unitaires par JUNIT

4.2. Test de validation

Avant la réalisation du codage, nous avons définis des cas d'utilisations.

Pour vérifier que le produit respecte les attentes du client, nous allons présenter ici les tests de validations qui s'appuieront sur les cas d'utilisations définis précédemment.

Les tests de validations sont calqués sur le modèle suivant :

Désignation	Démarche à suivre	Résultat attendu	Oui/non	Remarques
-	-	-	-	-

Tableau 7 – Modèle de test de Validation

Désignation	Démarche à suivre	Résultat attendu	Oui/non	Remarques
Gérer un institut	1 – Le gérant ajoute un institut 2 – Le gérant ajoute des départements à l'institut	On a un institut avec un ou plusieurs départements ainsi que un ou plusieurs étudiants.	Oui	-

	3 – Le gérant ajout des étudiants à l'institut			
--	------------------------------------------------	--	--	--

Tableau 8 – Test de validation #1

Désignation	Démarche à suivre	Résultat attendu	Oui/non	Remarques
Gérer un étudiant	1 – Le gérant ajoute un étudiant 2 – Le gérant effectue les modifications nécessaires 3 – Le gérant supprime un étudiant	L'étudiant est ajouté puis ses informations sont modifiées et à la fin l'étudiant n'existe plus	Oui	-

Tableau 9 – Test de validation #2

Désignation	Démarche à suivre	Résultat attendu	Oui/non	Remarques
Gérer un département	1 – Le gérant ajoute un département 2 – Le gérant effectue les modifications nécessaires 3 – Le gérant supprime un département	Le département est ajouté puis ses informations sont modifiées et à la fin le département n'existe plus	Oui	-

Tableau 10 – Test de validation #3

Désignation	Démarche à suivre	Résultat attendu	Oui/non	Remarques
Gérer un enseignant	1 – Le gérant ajoute un enseignant 2 – Le gérant effectue les modifications	L'enseignant est ajouté puis ses informations sont modifiées et à la fin l'enseignant n'existe plus	Oui	-

	nécessaires			
	3 – Le gérant supprime un enseignant			

Tableau 11 – Test de validation #4

Désignation	Démarche à suivre	Résultat attendu	Oui/non	Remarques
Gérer un cours	1 – Le gérant ajoute un cours 2 – Le gérant effectue les modifications nécessaires 3 – Le gérant supprime un cours	Le cours est ajouté puis ses informations sont modifiées et à la fin le cours n'existe plus.	Oui	-

Tableau 12 – Test de validation #5

4.3. Rapport d'erreur

Conclusion

5. Conclusion

Indexe

Figure 1 – Diagramme UML (bas niveau) du projet IUT	9
Tableau 1 – Modèle des Tests.....	12
Tableau 2 – Tests Unitaires de la classe Institut.....	13
Tableau 3 – Tests Unitaires de la classe Departement.....	13
Tableau 4 – Tests Unitaires de la classe Cours.....	13
Tableau 5 – Tests Unitaires de la classe Enseignant.....	14
Tableau 6 – Tests Unitaires de la classe Etudiant.....	14
Image 1 – Validation des tests unitaires par JUNIT.....	15
Tableau 7 – Modèle de test de Validation.....	15
Tableau 8 – Test de validation #1.....	16
Tableau 9 – Test de validation #2.....	16
Tableau 10 – Test de validation #3.....	16
Tableau 11 – Test de validation #4.....	17
Tableau 12 – Test de validation #5.....	17