
Dédicace

Je dédie ce travail :

À mes chers parents, à ma grande mère

Je n'ose pas imaginer une expression qui puisse résumer ma gratitude envers votre soutien et encouragement. Aucune dédicace ne saurait exprimer mon amour et mon respect pour vous.

Que ce travail soit pour vous un témoignage de ma vive reconnaissance et de mon amour filial.

À mon cher frère Nouaman et

Mes chères sœurs Siham, Hanane, et Bassma

Avec tout mon amour et mon affection, je vous souhaite une vie pleine de bonheur et de réussite et je vous remercie pour votre soutien durant toutes les périodes difficiles.

À mes amis, à l'EMI,

À tous ceux qui m'aiment

Sachez que vous m'êtes très chères et que je vous suis très reconnaissant pour tout l'amour et le soutien dont vous faites preuve à mon égard.

Je vous remercie du fond du cœur.

Anouar Mechti

Dédicace

*J'ai une dette particulière envers ma chère mère dont le soutien et la
sagesse m'ont guidé depuis ma naissance et me
guideront toujours.*

À mon cher père

Sans qui je ne serais jamais là où j'en suis.

À toute ma famille,

À mes très chers amis de l'EMI,

Pour tout le soutien que vous m'avez offert je vous dis MERCI.

À tous ceux qui m'aiment.

Otman IGHOULASSEN

Remerciements

Nous remercions tout d'abord Dieu tout puissant de nous avoir permis de mener à terme ce projet qui présente pour nous le point de départ de notre carrière.

Il nous est agréable d'exprimer notre reconnaissance auprès de toutes les personnes, dont l'intervention au cours de ce projet, a favorisé son aboutissement.

Ainsi, nous tenons à remercier nos encadrants **Mr. Khalil ATALLA**, **Mr. Abdelfattah Idri** et **Mr. Abdelhay MRABET** pour leurs directives précieuses et leurs conseils pertinents qui ont été d'un appui considérable au cours de ce projet.

Nous remercions également **Mlle Awatif BRIGUI** lauréat de l'EMI de nous avoir proposé ce projet et pour tous les conseils fructueux qu'elle n'a cessé de nous prodiguer. Nous tenons à remercier aussi toute l'équipe de Visual Soft pour leur accueil chaleureux.

La disponibilité, la gentillesse et la générosité de **M. Abderrahim HASBI** dont les précieux conseils et les orientations nous ont été d'un grand secours tout au long de ce stage. Qu'elle trouve ici, l'expression de notre profonde gratitude et de nos sincères remerciements.

Nous saisissons aussi l'occasion pour remercier tous les membres du jury, **M. Abdellah EL HADRI** et **M. Karim BOUZOUBAA** qui nous ont fait l'honneur d'accepter de juger ce travail.

Nos remerciements s'adressent également à tout le corps professoral de l'École Mohammadia d'Ingénieurs, pour la qualité de son enseignement ainsi que sa disponibilité tout au long de ces trois années de formation.

Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de nos remerciements les plus chaleureux.

Résumé

Notre projet de fin d'études effectué à VisualSoft consistait en l'analyse, la conception et la réalisation des modules de gestion des étudiants, des enseignants et des établissements de l'enseignement traditionnel marocain. Ces modules font partie, parmi d'autres, du grand projet à caractère national, lancé par le ministère des Habous et des affaires islamiques pour la mise en œuvre d'une solution intégrée de gestion de la scolarité dans les niveaux préscolaire, primaire, collégial, secondaire et terminal de l'enseignement traditionnel marocain.

Pour mener à bien notre travail, nous avons découpé le projet en quatre étapes. Au début, nous avons défini le contexte général du projet, et on a adopté comme démarche de conduite de projet, la méthode RUP (Rational Unified Process), qui est une méthode de développement logicielle itérative basée sur le langage UML. Ensuite, on est passé à la phase de l'analyse et la spécification des besoins. Dans l'étape suivante, nous avons abordé la phase de conception pour bien structurer la couche métier de notre application. Enfin, la dernière étape était la réalisation et la mise en œuvre des modules de notre système à l'aide des Frameworks techniques Java/JEE sous l'environnement de développement Netbeans.

Le présent rapport synthétise le travail que nous avons réalisé durant la période de notre stage, et se compose de quatre chapitres qui retracent les différentes étapes de notre projet.

Abstract

Our graduation internship performed at VisualSoft was about the analysis, design and implementation of the management modules of students, teachers and institutions of the Moroccan traditional teaching. These modules are part of the big national project launched by the Ministry of Endowments and Islamic Affairs for the implementation of an integrated solution for the schooling management in preschool, primary, secondary and terminal levels of the Moroccan traditional education.

Initially, we defined the general context of the project, and we adopted the Rational Unified Process (RUP) methodology as an approach to project management, which is an iterative software development methodology based on UML. Then we went to the phase of analysis and specification requirements. In the next step, we addressed the design phase to properly structure the business layer of our application. The final step of our project was the implementation of the modules of our system using Java/JEE technical frameworks under the Netbeans development tool.

This report presents the work that we perform during our internship, and consist of five chapters that trace the various stages of our project.

ملخص

يعرض هذا التقرير مختلف مراحل مشروع نهاية الدراسة الذي تم انجازه لدى شركة VisualSoft. وقد كان موضوع مشروعنا متعلقا بتحليل و تصميم وحدات تدبير الطلاب و الأساتذة ومؤسسات التعليم العتيق، و التي تندرج مع وحدات أخرى ضمن المشروع الوطني الذي أطلقته وزارة الأوقاف و الشؤون الإسلامية لإنجاز منظومة معلومية متكاملة لتدبير التمدريس على كل مستويات التعليم العتيق المغربي.

من أجل إنجاز عملنا على أكمل وجه، قمنا بتقسيم مهمتنا إلى أربع مراحل، حيث قمنا في البداية بتحديد المتطلبات الوظيفية و التقنية، و التي قمنا بتحليلها في مرحلة ثانية، لنقوم بعد ذلك بتصميم مختلف وحدات مشروعنا، مستعينين في ذلك بلغة النمذجة الموحدة UML.

أما المرحلة الأخيرة، فقد خصصت للدراسة التقنية و التطوير، وذلك بالاعتماد على إطارات عمل Java/JEE.

Liste des abréviations

Abréviation	Signification
AJAX	Asynchronous JavaScript + XML
API	Application Programming Interface
CRUD	Create Retrieve Update Delete
DAO	Data Access Object
DET	Département de l'enseignement traditionnel
DR	Délégation régionale
DP	Délégation provinciale
EET	Établissement de l'enseignement traditionnel
HTML	HyperText Mark-Up Language
IDE	Integrated Development Environment
IHM	Interface Homme Machine
JDBC	Java Database Connectivity
JEE	Java Entreprise Edition
JPQL	Java Persistence Query Language
JSF	Java Server Faces
JSTL	Java Standard Tag Library
MHAI	Ministère des Habous et des Affaires Islamiques
MVC	Model View Controller
ORM	Object-Relational Mapping
POJO	Plain Old Java Object
RUP	Rational Unified Process
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
UML	Unified Modeling Language
XML	Extensible Markup Language

Table des figures

Figure I-1 : Organigramme d'IT Group	4
Figure I-2 : Organigramme général de la Direction de l'Enseignement Traditionnel :	6
Figure I-3 Schéma de l'enchaînement des activités dans RUP	9
Figure II-1: Les acteurs du système.....	15
Figure II-2: Diagramme de contexte	16
Figure II-3: diagramme de packages des use case	17
Figure II-4: Cas d'utilisation « inscrire élève ».....	18
Figure II-5: Cas d'utilisation « Gérer établissement »	20
Figure II-6: Cas d'utilisation "gestion des cycles d'enseignement"	22
Figure II-7: Cas d'utilisation "gestion du personnel administratif et pédagogique"	24
Figure II-8: Diagramme d'activité "inscrire élève"	26
Figure II-9: Diagramme de séquence "ajouter établissement"	27
Figure II-10: Diagramme de séquence "inscrire élève"	28
Figure II-11: Diagramme de séquence "ajouter employé"	29
Figure III-1 Diagramme de paquetage système.....	32
Figure III-2: Diagramme d'interaction d'ajout d'un élève.....	34
Figure III-3: Diagramme de classes des informations personnelles de l'élève.....	35
Figure III-4: Diagramme d'interaction d'inscription d'un élève.....	36
Figure III-5: Diagramme de classes d'inscription.....	37
Figure III-6 : Diagramme de classes de parcours scolaire.....	38
Figure III-7 : Diagramme de classes du package "gestion des élèves et étudiants"	39
Figure III-8: Diagramme de classes de gestion des cycles, niveaux et classes	40
Figure III-9 : Diagramme de classes des locaux.....	41
Figure III-10: Diagramme de classes des employés et du personnel	42
Figure III-11: Diagramme de classes du package "gestion des établissements"	43
Figure IV-1: L'architecture technique de l'application	48
Figure IV-2: Interface de l'application: authentification	50
Figure IV-3: Interface de l'application: Menu principal.....	51
Figure IV-4: Interface de l'application: liste des élèves ajoutés	52
Figure IV-5: Interface de l'application: formulaire d'ajout d'un élève.....	53
Figure IV-6: Interface de l'application: afficher les informations un élève.....	53
Figure IV-7: Interface de l'application: Modifier un élève.....	54

Liste des tableaux

Tableau I-1: Les quatre phases du processus RUP.....	10
Tableau II-1: Description du cas d'utilisation « inscrire élève ».....	19
Tableau II-2: Description textuelle du cas d'utilisation « Gérer établissement »	21
Tableau II-3: Description textuelle « Gérer cycle d'enseignement ».....	23
Tableau II-4: Description textuelle du cas d'utilisation « gestion employés »	25

Table des matières

Introduction générale	1
I. Chapitre 1 : Contexte Général du Projet.....	3
1. Présentation de l'organisme d'accueil	3
2. Présentation générale du projet.....	5
2.1. Contexte organisationnel	5
2.2. Structure et statistiques	7
2.3. Objectifs du projet.....	7
2.4. Mission du stage.....	8
3. Conduite du projet : Méthode RUP (Rational Unified Process)	8
3.1. Motivation	8
3.2. Enchaînement d'activités essentielles dans RUP.....	9
3.3. Les quatre phases du processus RUP	9
4. Conclusion.....	10
II. Chapitre 2 : Étude fonctionnelle.....	12
1. Spécification des besoins	12
1.1. SPECIFICATION des besoins fonctionnels	12
1.2. Spécification des besoins techniques	13
2. Analyse du contexte	14
2.1. Les rôles	14
2.2. Diagramme de contexte	16
2.3. Diagramme des cas d'utilisation.....	16
2.4. Diagrammes d'activités	25
2.5. Diagrammes de séquences.....	27
3. Conclusion.....	30
III. Chapitre 3 : Conception	32
1. Diagramme de paquetage.....	32
2. Le package « Gestion des élèves et étudiants »	33
2.1. Diagramme d'interaction des informations personnelles de l'élève.....	33
2.2. Diagramme de classes des informations personnelles de l'élève	35
2.3. Diagramme d'interaction d'inscription.....	36
2.4. Diagramme de classes d'inscription	37
2.5. Diagramme de classes de parcours scolaire.....	38
3. Le package « Gestion des établissements »	40
3.1. Gestion des cycles, niveaux, et classes D'un établissement.....	40
3.2. La description des locaux.....	41
3.3. Définition des postes des cartes administratif, enseignants, et agents	42
3.4. Diagramme Global	42
4. Conclusion.....	44

IV. chapitre 4 : Mise en oeuvre.....	46
1. Le design pattern MVC.....	46
2. Java Enterprise Edition.....	46
3. Outils et Frameworks utilisés	46
3.1. Le framework JSF 2.0	46
3.2. Enterprise Java Beans (ejb 3).....	47
3.3. JPA 2.0	47
3.4. Le Framework EclipseLink.....	47
3.5. Oracle 10g.....	48
4. Architecture technique	48
5. Démarche de développement	49
6. Interfaces de l'application.....	50
6.1. Authentification.....	50
6.2. Gestion des étudiants	52
7. Conclusion.....	54
Conclusion générale	55
Bibliographie	56
Webographie.....	56
Annexe A : Outils utilisés	57
Annexe B : Le cycle de vie à phases multiples d'une page JSF	59
Annexe C : Les maquettes	62
Annexe D : Diagramme de classes global	64

Introduction générale

La stratégie du développement économique et social qui est adoptée par le gouvernement marocain depuis le début de ce troisième millénaire s'articule autour de plusieurs axes, notamment la réforme et la modernisation de l'administration publique, et cela à travers la promotion de l'utilisation des technologies de l'information et de la communication dans les différents départements ministériels.

Dans ce cadre, le ministère des Habous et des affaires islamiques a passé à VisualSoft, notre société d'accueil, un projet à caractère national, pour la mise en œuvre d'une solution intégrée de gestion de la scolarité dans les niveaux préscolaire, primaire, collégial, secondaire et terminal de l'enseignement traditionnel marocain (SIGES).

Notre mission dans ce stage au sein de VisualSoft, consiste en l'analyse, la conception et la réalisation des modules de gestion des établissements, des élèves et des enseignants.

Le présent rapport se compose de quatre chapitres. Le premier définit le contexte général du projet, à savoir la présentation de l'organisme d'accueil et la description de notre projet.

Le deuxième chapitre présente une étude fonctionnelle dont l'objectif est de capturer les besoins fonctionnels du système futur ainsi que de faire son analyse.

Le troisième chapitre s'intéresse à la conception des différents modules de notre projet moyennant les diagrammes de classes UML.

Nous nous penchons ensuite dans le quatrième et dernier chapitre sur l'étude technique et la mise en œuvre de notre projet. En effet, on va décrire dans un premier temps l'architecture technique, les outils, et les Frameworks utilisés pour le développement. Ensuite, nous allons expliquer la démarche de développement et illustrer certaines fonctionnalités assurées par notre application à travers quelques écrans.

Finalement, une conclusion récapitule le travail accompli dans le cadre de ce projet de fin d'études et présente des axes d'amélioration susceptibles de raffiner davantage l'outil développé.

Chapitre 1 : Contexte Général du Projet

Ce chapitre présente le contexte, la motivation et les objectifs de notre projet de fin d'études. Nous décrivons dans un premier temps l'organisme d'accueil. Nous présentons ensuite le cadre général du projet et la mission qui nous a été confiée, et en fin nous abordons la démarche et la conduite du projet.

I. CHAPITRE 1 : CONTEXTE GENERAL DU PROJET

1. Présentation de l'organisme d'accueil

IT Group Morocco est un groupement composé de cinq filiales indépendantes et complémentaires dans leur fonctionnement : *I.T. Consulting, RSI, VisualSoft, Business Apps et IT académie*, spécialisé en étude et conseil en Systèmes d'Information et en développement d'applications client/serveur, Intranet/Internet et formation.

Ce rapprochement qu'offre IT Group entre la gouvernance des systèmes d'information, l'expertise en connectivités S.I. et l'Intégration de technologies Web permet de proposer sous une seule entité une offre complète aux clients, IT Group Morocco devient ainsi un grand Groupe marocain dans le monde des Systèmes d'Information doté de nombreuses et prestigieuses références.

Nous avons effectué notre stage de fin d'études au sein de la filiale *Visual Soft*. C'est une société de services en ingénierie informatiques (SSII) créée en 1997, et qui a pour principal objectif de fournir, aux entreprises et organismes divers, des produits et des services professionnels de haute qualité.

Visual Soft entretient une culture d'entreprise centrée sur la qualité et sur le contentement et la satisfaction du client. Se faisant, elle s'engage à entretenir avec sa clientèle des rapports sérieux et permanents fondés sur des qualités de compétence et de professionnalisme.

Pour répondre au mieux aux exigences de ses clients, et mener à bien sa mission, *Visual Soft* réunit plusieurs domaines de compétences :

- Le conseil et l'assistance en informatique, et l'étude et la mise en place de systèmes d'informations ;
- Le développement des applications. Ces dernières sont, soit des produits standards, soit des développements spécifiques aux besoins et à la demande des clients. Les produits standards sont destinés aux entreprises privées (gestion de pharmacies, de cabinets dentaires, de laboratoires d'analyses médicales, de la comptabilité...) et aux organismes publics (gestion des marchés, du personnel, du stock...).

- Le développement de sites Web, ainsi que la conception et la mise en place de serveurs Intranet et Internet pour la publication de données et d'informations ou pour des applications Client-Serveur.
- La formation aux techniques et outils de développement, aux systèmes d'exploitation et aux réseaux, et l'utilisation avancée des logiciels de Bureautique.

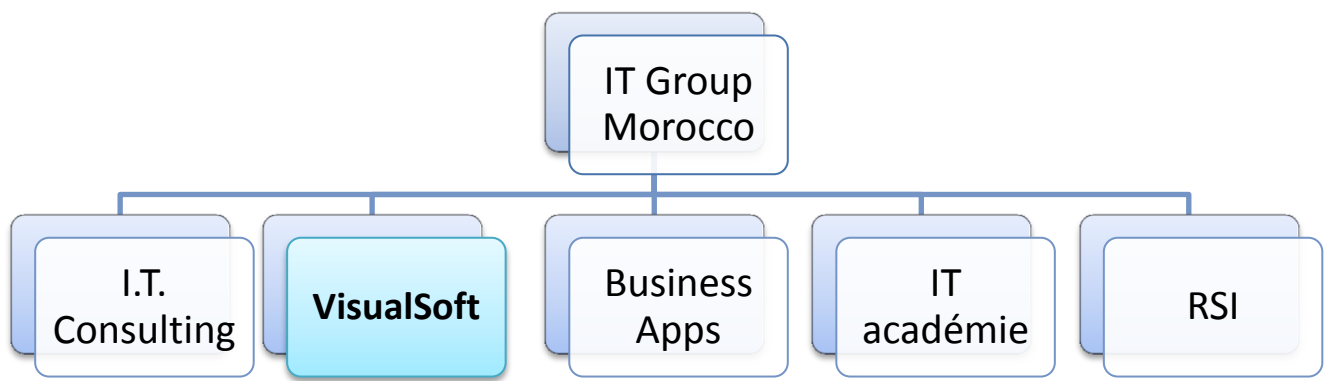


Figure I-1 : Organigramme d'IT Group

2. Présentation générale du projet

Dans le cadre de l'informatisation de ses différents services et activités, le Ministère des Habous et des Affaires Islamiques a passé à Visual Soft un projet à caractère national, pour la mise en œuvre d'une solution intégrée de gestion de la scolarité (SIGES) dans les niveaux préscolaire, primaire, collégial, secondaire et terminal de l'enseignement traditionnel marocain.

L'approche retenue pour la réalisation de ce projet de grande envergure, exige une solution cible qui soit performante, standardisée au niveau national, touchant les différents niveaux de gouvernance (établissements scolaires, délégations et services centraux).

2.1. Contexte organisationnel

Les établissements de l'enseignement traditionnel, publics ou privés, déclarés auprès du Ministère des Habous et des Affaires Islamiques, sont gérés par:

- La Direction de l'Enseignement Traditionnel pour les activités de consolidation, suivi, validation, octroi des autorisations, production des situations statistiques, contrôle... ;
- Les délégations régionales et provinciales des Affaires Islamiques (qui sont au nombre de 82) pour la collecte des informations émanant des établissements, la coordination, le suivi ...;

Toutes les informations locales de gestion des établissements, sont partagées avec les niveaux supérieurs: provincial, régional et central.

La Direction de l'Enseignement Traditionnel a notamment pour mission de :

- Fixer la stratégie pédagogique relative à l'enseignement traditionnel;
- Établir les régimes des études et des examens ;
- Délivrer les autorisations d'ouverture des établissements de l'enseignement traditionnel et veiller à l'application des textes législatifs et réglementaires y afférant, notamment en ce qui concerne le contrôle du respect des obligations pédagogiques et administratives ;
- Examiner les demandes et les propositions de transfert des établissements privés d'enseignement traditionnel ;
- Proposer les mesures de soutien et d'assistance aux établissements de l'enseignement traditionnel afin d'accomplir leur mission dans de bonnes conditions;

- Contrôler et évaluer l'exécution des programmes et des méthodes ;
- Veiller à la formation initiale et à la formation continue des cadres de l'enseignement traditionnel ;
- Créer des espaces d'éducation et des activités parascolaires en faveur des élèves et étudiants de l'enseignement traditionnel ;
- Assurer la coordination et le contact avec les autorités concernées par les questions de l'enseignement et de la formation.

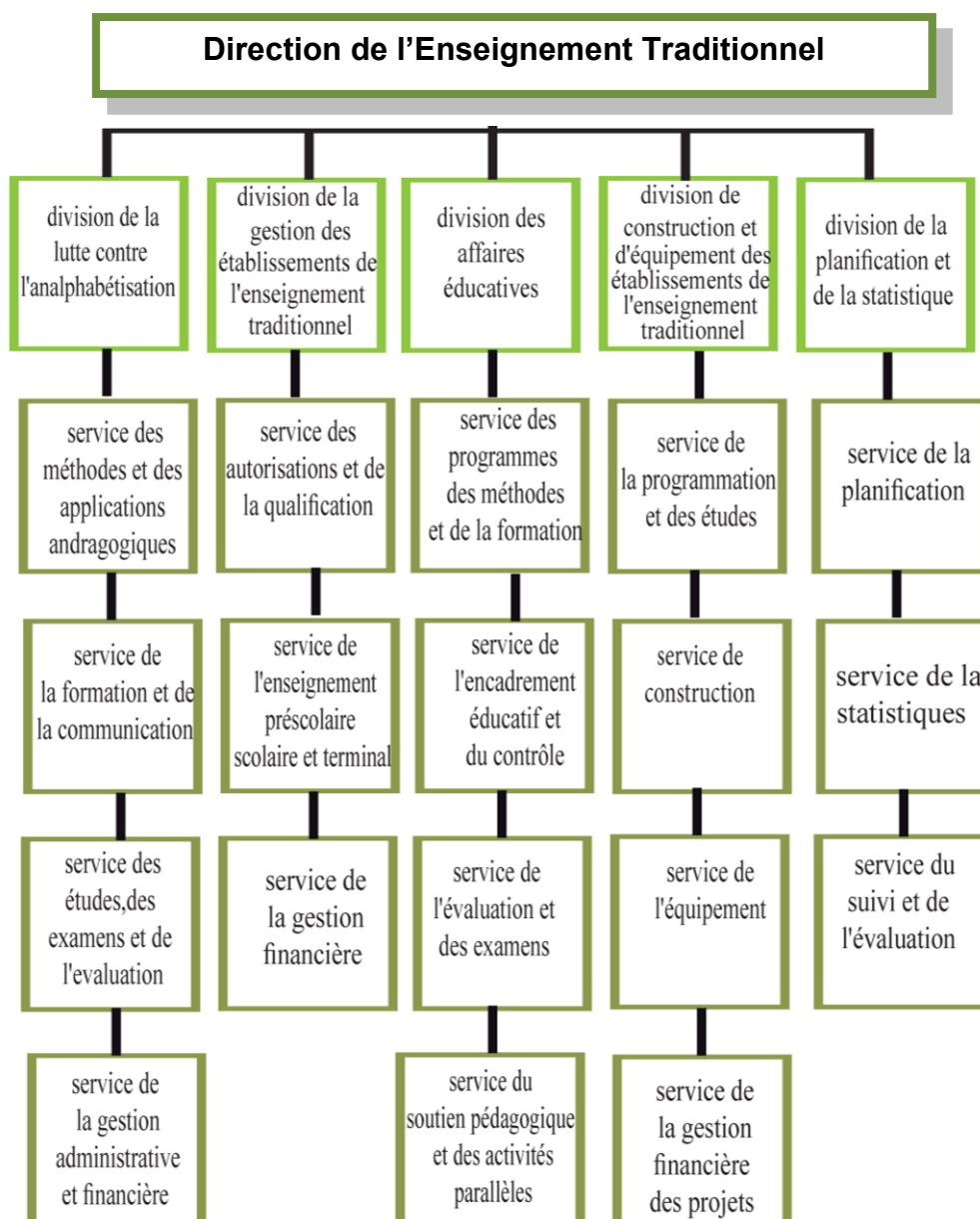


Figure I-2 : Organigramme général de la Direction de l'Enseignement Traditionnel :

2.2. Structure et statistiques

Les établissements de l'enseignement traditionnel sont classés en quatre cycles d'enseignement, à savoir :

- Le préscolaire, où le cursus normal est de 2 ans (les établissements préscolaires sont les écoles coraniques –*katatib*- كَتَاتِيب قرآنية) ;
- Le primaire, où le cursus normal est de 6 ans ;
- Le collégial, où le cursus normal est de 3 ans ;
- Le secondaire, où le cursus normal est de 3 ans ;
- Le terminal, où le cursus normal est de 3 ans ;

Les établissements primaires, collégiaux, secondaires et terminaux, sont environ au nombre de 463, Ils comptent approximativement 26000 élèves, 4000 enseignants et près de 1000 employés (corps administratif et agents). Les *katatib* comptent près de 330 000 élèves encadrés par environ 13 000 enseignants (à raison – généralement- d'un enseignant par école préscolaire).

2.3. Objectifs du projet

L'objectif du projet est la réalisation et la mise en place d'une solution intégrée de gestion de la scolarité (SIGES), qui intègre notamment les modules suivant :

- Gestion des établissements ;
- Gestion des centres d'examen ;
- Gestion des enseignants, du personnel de l'administration scolaire des établissements, et des cadres pédagogiques ;
- Gestion des élèves et étudiants ;
- Inspections administrative et pédagogique ;
- Calendrier scolaire ;
- Programmes et manuels scolaires ;
- Horaire et emploi du temps ;
- Soutien pédagogique ;
- Examens ;
- Absences et retards.

La solution intégrée cible doit permettre notamment :

- L'automatisation de l'ensemble des procédures de gestion ;

- La consolidation des données dans une seule base de données nationale pour notamment suivre le parcours des élèves et étudiants ;
- La reprise de l'existant ;
- L'élaboration des statistiques, des éditions et des consultations multicritères ;
- L'archivage des données.

2.4. Mission du stage

Visual Soft a estimé la durée totale pour la réalisation de ce projet à 900jrs/homme. De ce fait notre chef de projet, et vue la durée de notre stage, nous a confié comme mission durant notre stage l'analyse, la conception et la réalisation des modules suivants :

- Gestion des établissements ;
- Gestion des élèves/étudiants ;
- Gestion des enseignants, du personnel de l'administration scolaire des établissements et des cadres pédagogiques ;

3. Conduite du projet : Méthode RUP (Rational Unified Process)

3.1. Motivation

Au départ, Visual Soft ne savait pas toutes les exigences du projet et comment les spécifier de façon précise. Nous avons donc senti que nous serons confrontés à des changements continuels, et donc il nous a fallu suivre une démarche itérative. Aussi, il fallait présenter des livrables, et donc la démarche devrait être incrémentale (c'est-à-dire qu'à la fin de chaque itération, une version exécutable est livrée). Et enfin, la méthode choisie devrait être construite sur UML et surtout conduite par les cas d'utilisation.

Pour toutes ces raisons, nous avons opté pour la méthode RUP (Rational Unified Process) qui est un processus de conception/développement de logiciel défini par Rational Software, unifié, c'est-à-dire construit sur UML, itératif, incrémental, centré sur l'architecture et conduit par les cas d'utilisation.

3.2. Enchaînement d'activités essentielles dans RUP

La méthode RUP propose le cycle de conception/développement suivant (Figure I-3) :

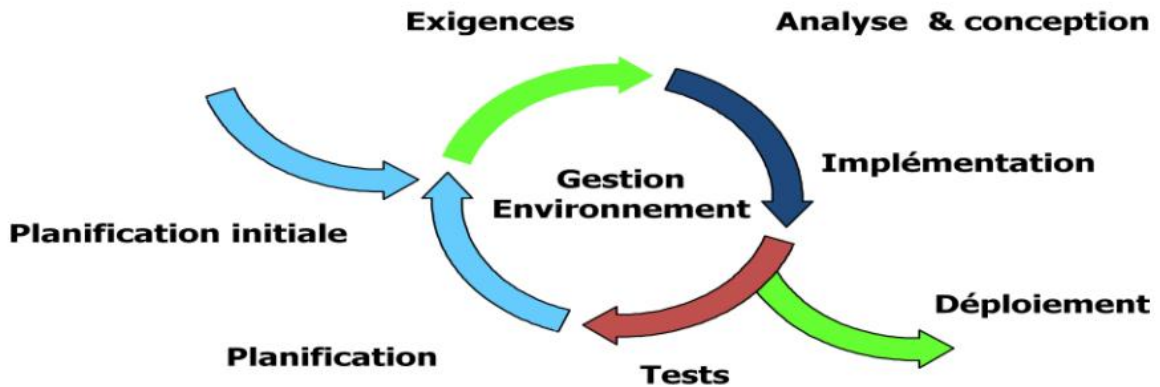


Figure I-3 Schéma de l'enchaînement des activités dans RUP

Nous avons veillé à le respecter tout le long de la réalisation de notre projet. En effet, nous avons commencé par une planification initiale en termes de temps et de ressources.

Ensuite, nous sommes passés à la définition des exigences qui a pour but de définir une vision du produit, et de traduire cette vision en cas d'utilisation

L'étape d'analyse nous a permis de répondre à la question « Quoi faire ? » Tandis que la conception à « Comment le faire ? ».

Nous avons ensuite implémenté l'application, tout en procédant par des tests unitaires afin d'évaluer le niveau de qualité atteint, pour enfin les déployer.

3.3. Les quatre phases du processus RUP

La méthode RUP est composée de quatre phases principales comme il est illustré dans le tableau suivant :

Phase	Objectifs
Inception	<ul style="list-style-type: none">- Comprendre le périmètre du projet.- Définir les objectifs du projet- Identification des acteurs et des cas d'utilisation- Choix des outils de développement.
Élaboration	<ul style="list-style-type: none">- Spécifier les exigences,- Description détaillée des cas d'utilisation- Conception des composants du système- Implémentation de quelques scénarios critiques
Construction	<ul style="list-style-type: none">- Finaliser l'analyse, conception, implémentation, et test des cas d'utilisations.- Une première version fonctionnelle est livrée.
Transition	<ul style="list-style-type: none">- Préparation de l'environnement de déploiement ;- Test du produit et correction d'erreurs en vue de livraison ;- Production de documentation

Tableau I-1: Les quatre phases du processus RUP

4. Conclusion

Dans ce chapitre, nous avons commencé par une présentation de l'organisme d'accueil. Nous sommes ensuite passés à la description du contexte, les motivations les objectifs du projet, ainsi que la mission de notre stage. On a ensuite présenté le processus de développement RUP, que nous avons adopté pour conduire notre travail.

Dans ce qui suit, nous allons nous intéresser à l'analyse et la conception et à l'élaboration des différents diagrammes UML.

Chapitre 2 : Étude fonctionnelle

Ce chapitre traite les deux phases les plus cruciales de tout cycle de développement, à savoir la capture des besoins fonctionnels et l'analyse. La première partie de ce chapitre présente la phase de spécification des besoins fonctionnels et techniques pour identifier les fonctionnalités auxquels le système doit répondre. Les résultats de la phase de spécification seront à la base de la phase d'analyse traitée dans la deuxième partie.

II. CHAPITRE 2 : ÉTUDE FONCTIONNELLE

1. Spécification des besoins

1.1. SPECIFICATION des besoins fonctionnels

1.1.1. Étude fonctionnelle du module : Gestions des élèves et des étudiants

La solution ciblée envisagée doit assurer toutes les opérations de gestion en vigueur sur une entité élève/étudiant, notamment :

- La création d'un nouvel étudiant (première inscription),
- La réinscription d'un étudiant,
- Le transfert d'inscription d'un établissement à un autre,
- La gestion des absences et d'assiduité,
- La gestion des différents cycles et niveaux scolaires, ainsi que la gestion des classes.

L'application doit aussi permettre de garder toutes les informations pertinentes qui contiennent les données personnelles des étudiants (date et lieu de naissance, adresse, numéro de téléphone, nom et prénom de l'élève et de ses parents ou tuteurs ...), ainsi que les informations concernant le parcours scolaire de l'élève (les établissements et les classes fréquentées, les notes, ...).

1.1.2. Étude fonctionnelle du module : Gestions des établissements

La gestion des établissements de l'enseignement traditionnel doit permettre d'assurer toutes les opérations en vigueur sur une entité établissement, notamment :

- la création d'un nouvel établissement,
- Fermeture d'un établissement.
- Les différents cycles scolaires autorisés, possibilité de changement de cycle.

L'application cible doit aussi être en mesure de détenir et sauvegarder des informations concernant l'établissement (nom, adresse, localisation, milieu (urbain ou

rural), statut (public ou privé), description des locaux (salles de classes, salle de bureaux, internats, réfectoires, cuisines, magasins, bibliothèques)).

Elle doit aussi permettre la définition des postes des cadres administratifs, des enseignants et des agents, ainsi que leurs fonctions et occupations, elle doit aussi permettre de générer et d'imprimer des rapports, autorisations et des statistiques.

1.2. Spécification des besoins techniques

✓ Ergonomie :

L'ergonomie de l'application doit favoriser la lisibilité et l'accessibilité des informations, ainsi que leur réutilisation.

L'interface doit être conviviale, facile à manipuler par des utilisateurs simples (non informaticiens).

✓ Bilinguisme :

Le système doit prendre en charge les données en arabe et en français simultanément, au moins pour les noms d'élèves et personnel, pour toutes les nomenclatures de base.

Le système doit prendre en charge les données en arabe ou en français, selon le choix de l'opérateur. L'interface utilisateurs doit être en langue arabe et française.

✓ Administration :

Le système doit permettre aux administrateurs de gérer les utilisateurs et groupes, leurs rôles et fonctions par module, et également de surveiller et gérer la liste des utilisateurs en ligne.

✓ Sécurité et intégrité :

- La solution doit avoir une interface d'authentification unique.
- Tous les modules font partie de la même solution avec le principe de saisie unique.
- L'accès aux fonctions du système et aux données sera permis seulement à base d'authentifications par nom d'utilisateur et mot de passe.
- Les utilisateurs, les groupes d'utilisateurs et les fonctions doivent être gérés par les administrateurs à l'aide des interfaces appropriées.

2. Analyse du contexte

L'analyse du contexte a pour objectif l'identification des différents acteurs qui interagiront avec le système. Ces acteurs seront regroupés par la suite sous forme de rôles dont chacun englobe un ensemble d'entités. Le regroupement sera effectué en se basant sur les interactions.

2.1. Les rôles

Un rôle réfère à une mission et à des responsabilités assumées par les acteurs du système étudié. Un acteur (utilisateur, dispositif matériel ou autre système) interagit directement avec le système et peut consulter et/ou modifier directement l'état du système, par émission et/ou réception de messages éventuellement porteurs de données.

L'étude de la typologie des rôles a conduit à la classification suivante :

- Au niveau central :
 - ✓ La Direction de l'Enseignement Traditionnel : Gestionnaires
- Au niveau régional et provincial :
 - ✓ Les délégations régionales du MHAI : Gestionnaires
 - ✓ Les délégations provinciales du MHAI : Gestionnaires
 - ✓ Les Établissements de l'Enseignement Traditionnel : Gestionnaires, Enseignants, Étudiants.

Les acteurs que nous avons identifiés à partir de la description des fonctionnalités ci-dessus sont les suivants :

- Gestionnaires :
 - ✓ Gestionnaire DET : les différentes opérations de gestion sur les établissements au niveau national (création, modification, fermeture, ...)
 - ✓ Gestionnaire DR : les différentes opérations de gestion au niveau régional, notamment la gestion des transferts des étudiants et des enseignants, et l'attribution de bourses.
 - ✓ Gestionnaire DP : les différentes opérations de gestion au niveau régional, la gestion des transferts, la gestion des candidatures libres aux examens.
 - ✓ Gestionnaire EET : gestion des inscriptions des étudiants, des absences et retards.
- Utilisateurs externes :

- ✓ Enseignant : Consultation des listes des étudiants, éditions des notes.
- ✓ Étudiant : Consultation des notes, des informations personnelles.
- Administrateurs :
 - ✓ Administration du système et gestion des rôles et des utilisateurs.
- Utilisateurs simples (internaute) : ils ont seulement le droit de consulter les données du système et de faire des recherches suivant des critères précis.

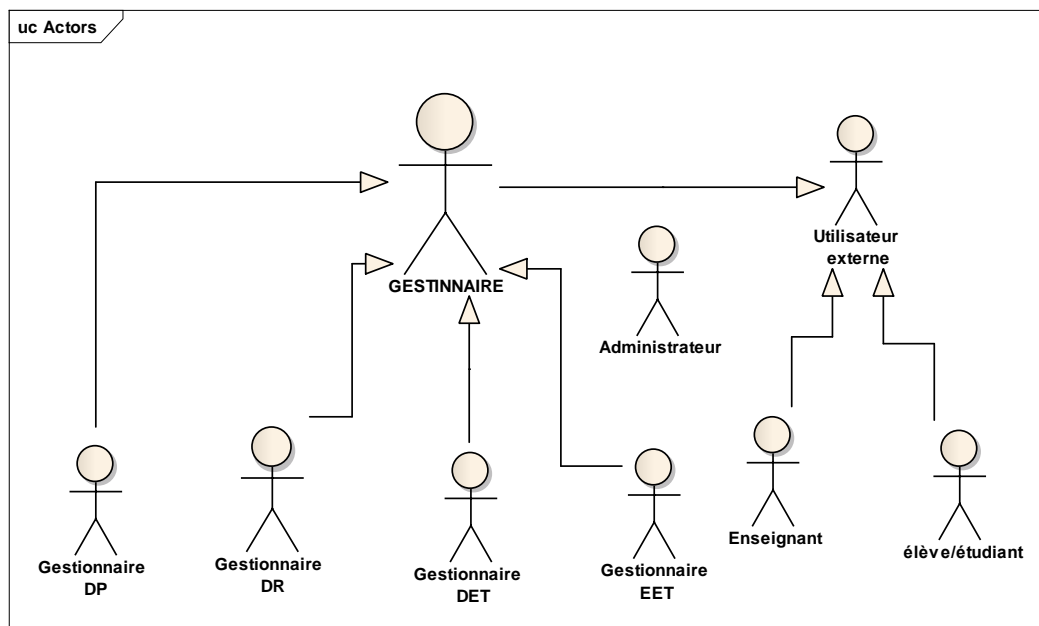


Figure II-1: Les acteurs du système

2.2. Diagramme de contexte

Le diagramme de contexte a pour but de représenter les flux d'informations entre le système et les acteurs externes. Il permet d'isoler le système en le délimitant et ainsi définir les éléments externes avec lesquels il interagit. C'est ainsi que l'on modélise le contexte général du projet par le diagramme de contexte ci-dessous (Figure II-2):

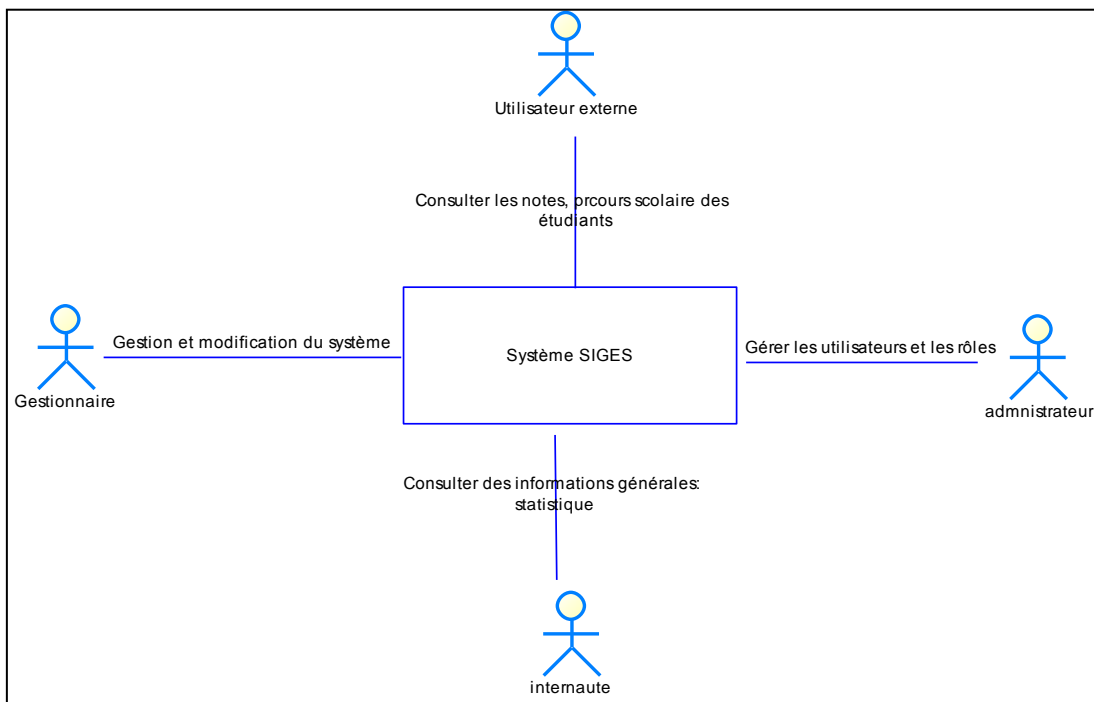


Figure II-2: Diagramme de contexte

2.3. Diagramme des cas d'utilisation

Les cas d'utilisation permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système.

Dans cette section nous allons présenter les cas d'utilisation concernant les modules de :

- ✓ gestion des étudiants,
- ✓ gestion des établissements,
- ✓ gestion des enseignants, du personnel de l'administration scolaire des établissements et des cadres pédagogiques,

Les acteurs principaux de notre système sont au nombre de quatre : Administrateur, Gestionnaires, utilisateurs externes (enseignants ou étudiants) et l'utilisateur simple (internaute).

Le diagramme ci-dessous (Figure II-3) regroupe les différents cas d'utilisations et les acteurs de notre système sous forme de packages :

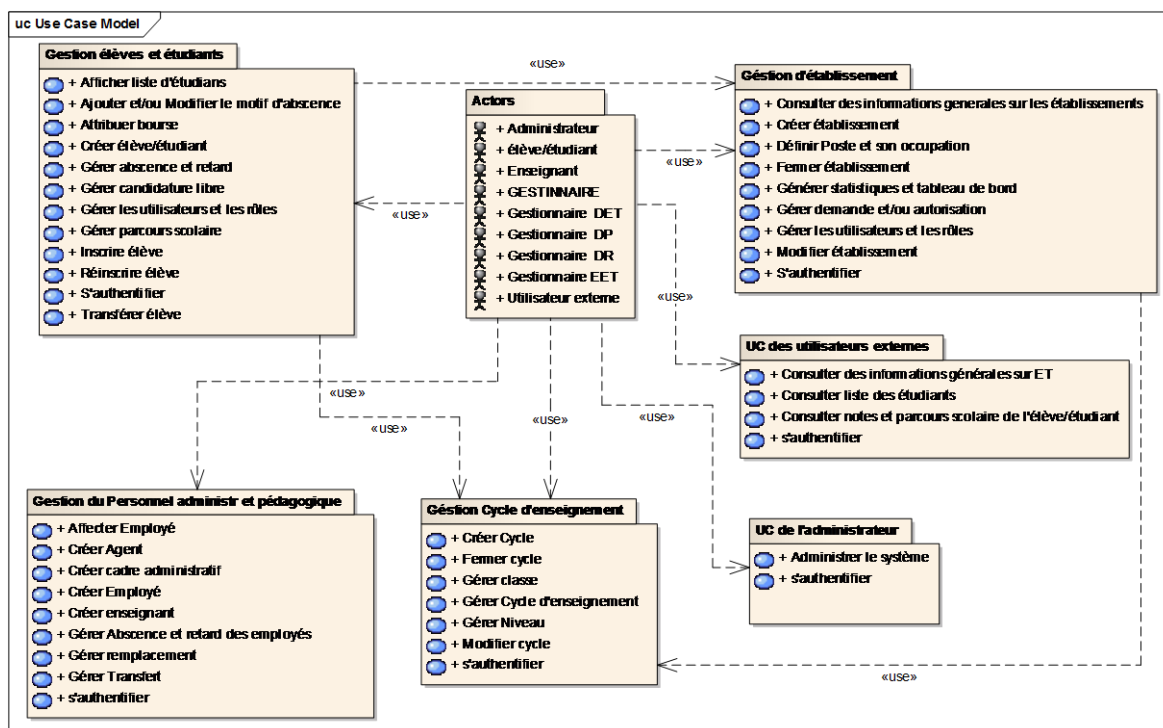


Figure II-3: digramme de packages des use case

2.3.1. Gestion des élèves et des étudiants

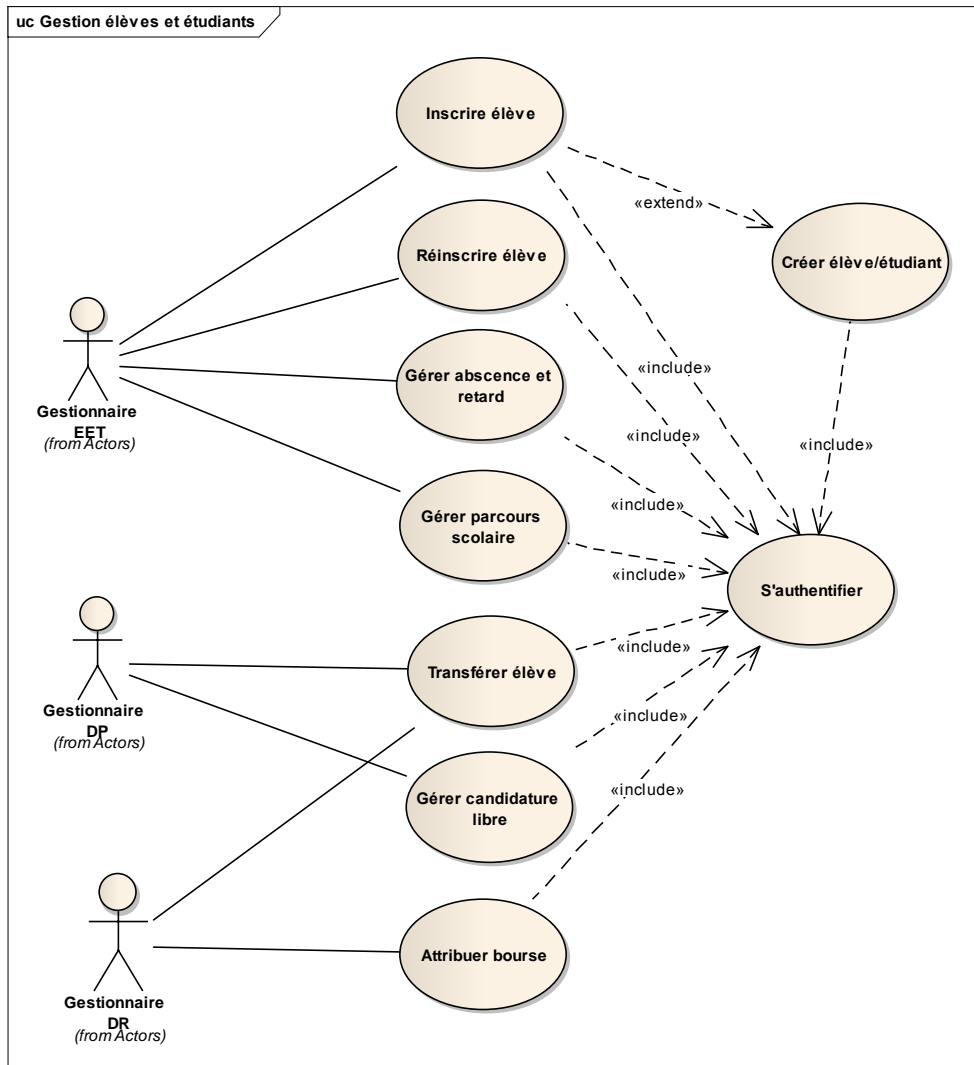


Figure II-4: Cas d'utilisation « inscrire élève »

➤ Description du cas d'utilisation « inscrire élève »

Cas d'utilisation	Inscrire élève
Version	Version 1.0 du 26/03/2012
Acteur principal	- Gestionnaire EET (La scolarité)
Objectifs	- Ce cas d'utilisation permet d'inscrire un élève ou étudiant dans un établissement d'enseignement traditionnel.
Pré conditions	- S'authentifier. - Vérification préalable des conditions d'inscription (âge, compétences...)
Post conditions	- Un nouvel élève/étudiant est inscrit, le nombre d'élèves/étudiants s'incrémente
Scénario nominal	- L'élève/étudiant présente sa demande, les informations nécessaires et les frais d'inscription - Gestionnaire EET vérifie que toutes les informations sont correctes. - Gestionnaire EET remplit le formulaire et enregistre le nouvel élève/étudiant, l'élève/étudiant est alors inscrit.
Scénario secondaire	L'EET annule l'inscription : Le système n'enregistre rien et le cas d'utilisation se termine en échec L'âge de l'élève n'est pas valide : Le système signale l'erreur mais peut continuer l'opération d'inscription
Exigences supplémentaires	L'inscription doit être la plus rapide possible ; Le système doit permettre de numériser la photo de l'élève/étudiant ; Le système doit supporter la saisie de quelques informations en arabe.

Tableau II-1: Description du cas d'utilisation « inscrire élève »

2.3.2. Gestion des établissements

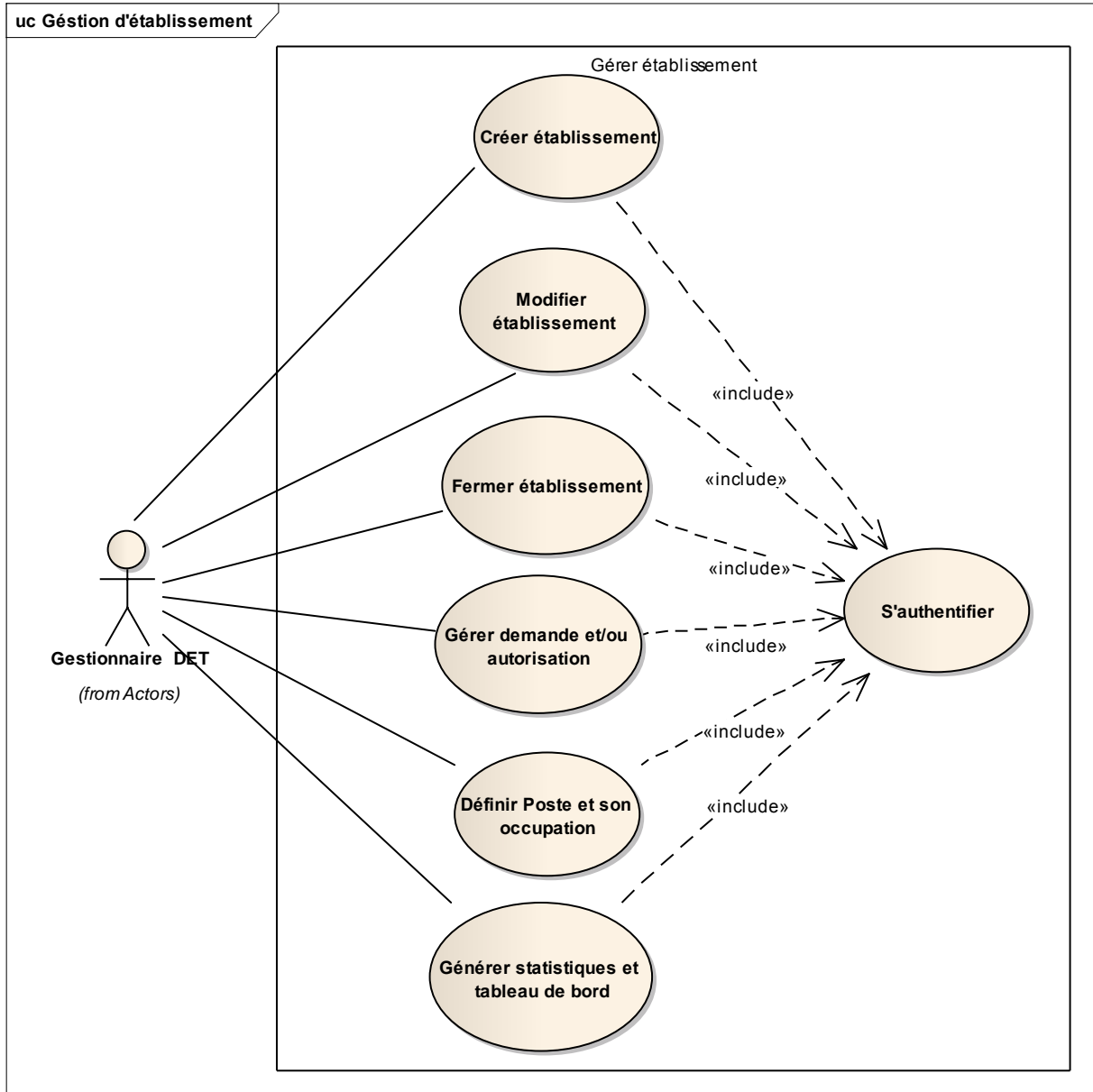


Figure II-5: Cas d'utilisation « Gérer établissement »

➤ Description textuelle du cas d'utilisation « Créer établissement » :

Cas d'utilisation	Gérer établissement
Version	Version 1.0 du 26/03/2012
Acteur principal	- Gestionnaire DET
Objectifs	- Ce cas d'utilisation permet de créer un nouvel établissement de l'enseignement traditionnel et de l'ajouter dans la base de données.
Pré conditions	- L'établissement a été inauguré ou autorisée par la DET. - S'authentifier
Post conditions	- Un nouvel établissement d'enseignement est créé et ajouté à la base de données.
Scénario nominal	- Saisir les données concernant l'établissement : le gestionnaire DET ouvre le formulaire d'ajout d'un établissement, remplit les informations nécessaires le concernant et valide l'ajout.
Scénario secondaire	- Consulter la liste des établissements - Modifier un établissement : s'il s'aperçoit qu'une information est erronée, le gestionnaire DET peut le modifier

Tableau II-2: Description textuelle du cas d'utilisation « Gérer établissement »

2.3.3. Gestion des cycles d'enseignement

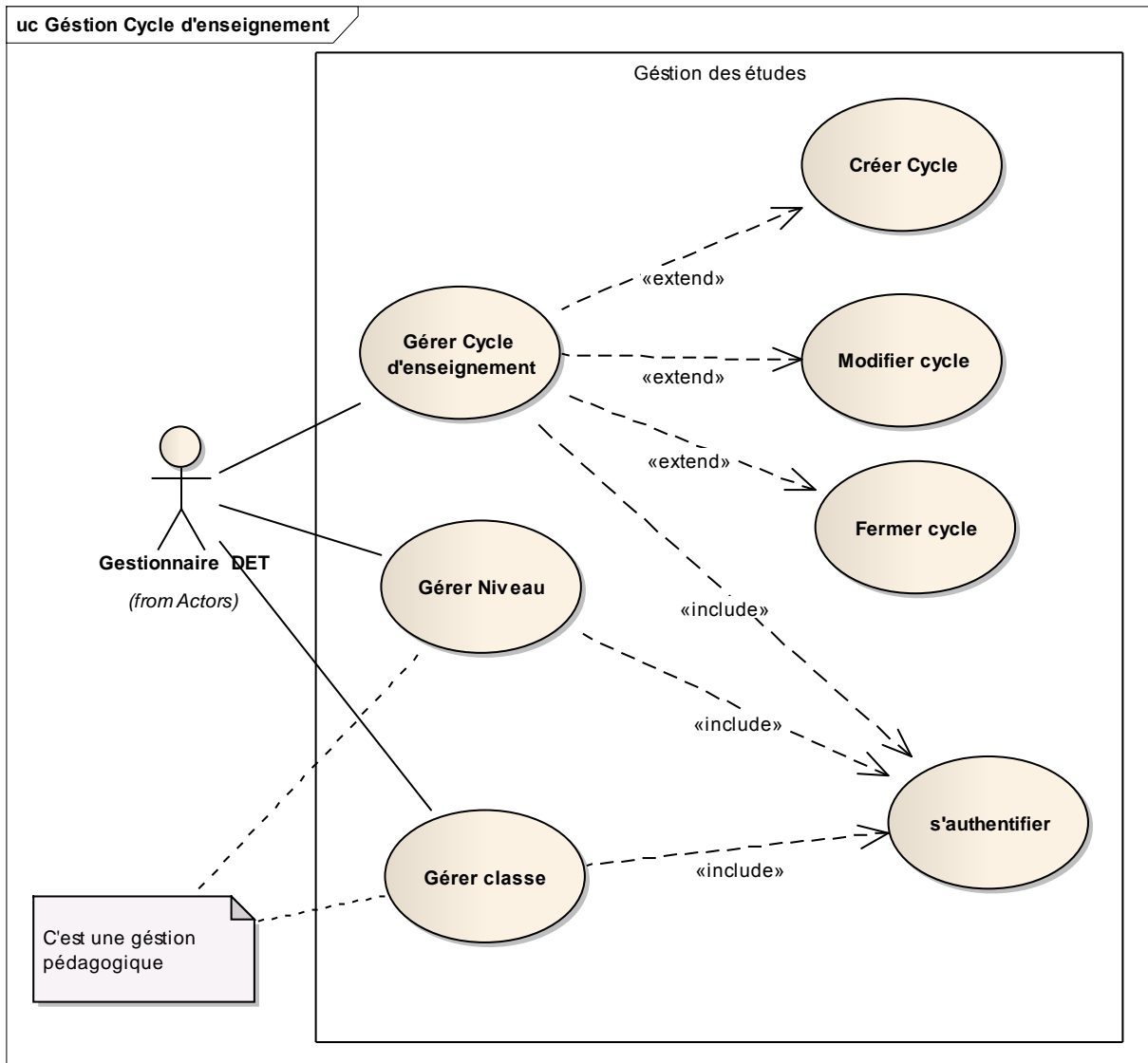


Figure II-6: Cas d'utilisation "gestion des cycles d'enseignement"

➤ Description textuelle « Gérer cycle d'enseignement »

Cas d'utilisation	Gérer un cycle d'enseignement
Version	Version 1.0 du 26/03/2012
Acteur principal	Gestionnaire DET
Objectifs	Ce cas d'utilisation permet de gérer un cycle d'enseignement concernant un établissement : <ul style="list-style-type: none">- Autorisation d'un cycle- Création d'un nouveau cycle- Changement de cycle
Pré conditions	- S'authentifier
Scénario nominal	- choisi un établissement, et lui attribue un cycle.
Scénario secondaire	- Le gestionnaire modifie une information erronée concernant un cycle - Consultation des cycles autorisés dans un établissement

Tableau II-3: Description textuelle « Gérer cycle d'enseignement »

2.3.4. Gestion du personnel administratif et pédagogique

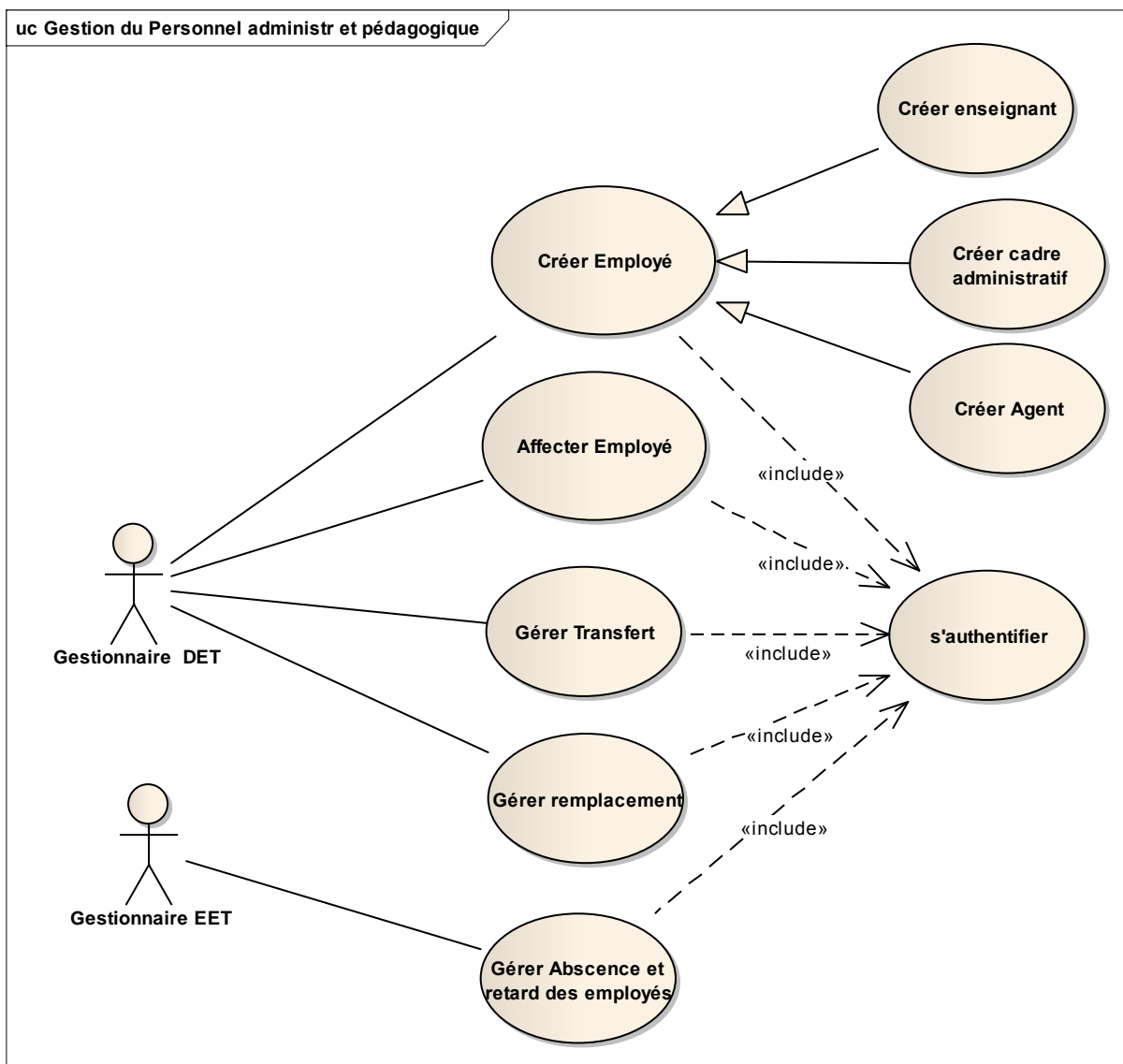


Figure II-7: Cas d'utilisation "gestion du personnel administratif et pédagogique"

Description textuelle du cas d'utilisation « gestion employés »

Cas d'utilisation	Créer employé
Acteur principal	Gestionnaire DET
Objectifs	Ce cas d'utilisation permet de saisir les informations d'un nouvel employé
Pré conditions	Employé ayant été recruté par la DET. S'authentifier
Post conditions	Nouvel employé crée et ajouté à la base de données

Scénario nominal	le gestionnaire DET ouvre le formulaire d'ajout d'un employé, remplit le formulaire d'ajout et le valide.
Scénario secondaire	Le gestionnaire consulte les informations d'un employé, les modifie en cas de données erronées.

Tableau II-4: Description textuelle du cas d'utilisation « gestion employés »

2.4. Diagrammes d'activités

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Nous allons présenter dans cette section les diagrammes d'activités concernant le processus d'inscription d'un étudiant, et celui de la gestion d'absence au sein d'un établissement d'enseignement.

2.4.1. Diagramme d'activité d'inscription d'étudiants

En effet, après la demande d'inscription de l'étudiant, le gestionnaire EET vérifie si l'étudiant vérifie les conditions d'admission, et l'ajoute à la base de données et l'enregistre en lui affectant l'établissement, le cycle et la classe.

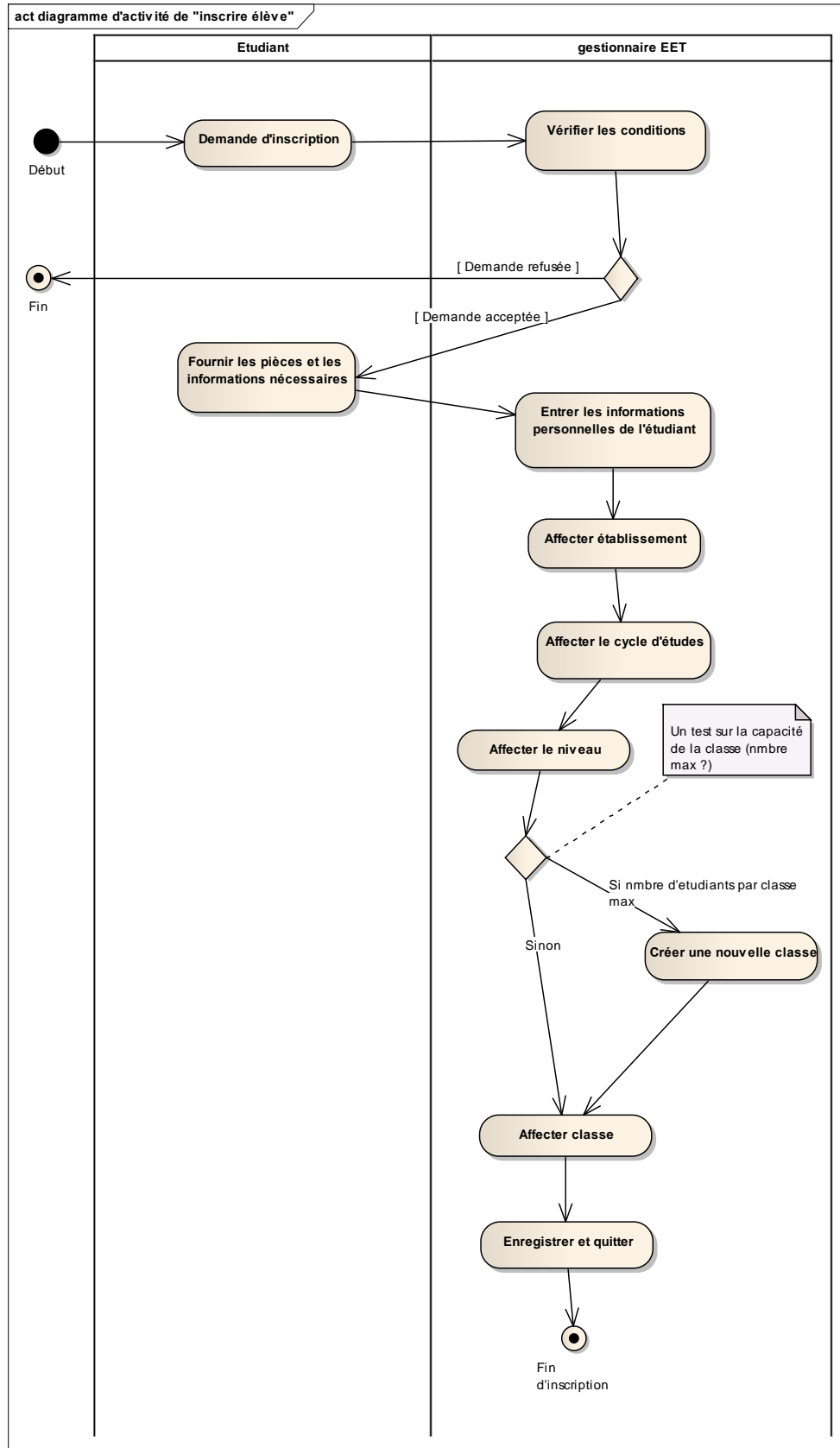


Figure II-8: Diagramme d'activité "inscrire élève"

2.5. Digrammes de séquences

Les diagrammes de séquence sont des diagrammes d'interactions qui permettent de modéliser les scénarios. Ils ont pour objectif de mieux représenter les interactions entre les objets de notre projet selon un point de vue temporel.

En effet, une fois les besoins du système formalisés à travers les diagrammes de cas d'utilisation, il faut ensuite modéliser les interactions entre le système et les différents acteurs à travers la succession séquentielle des actions.

Dans cette section nous allons présenter des exemples de diagrammes de séquences de notre système. Diagramme de séquence d'ajout d'un établissement :

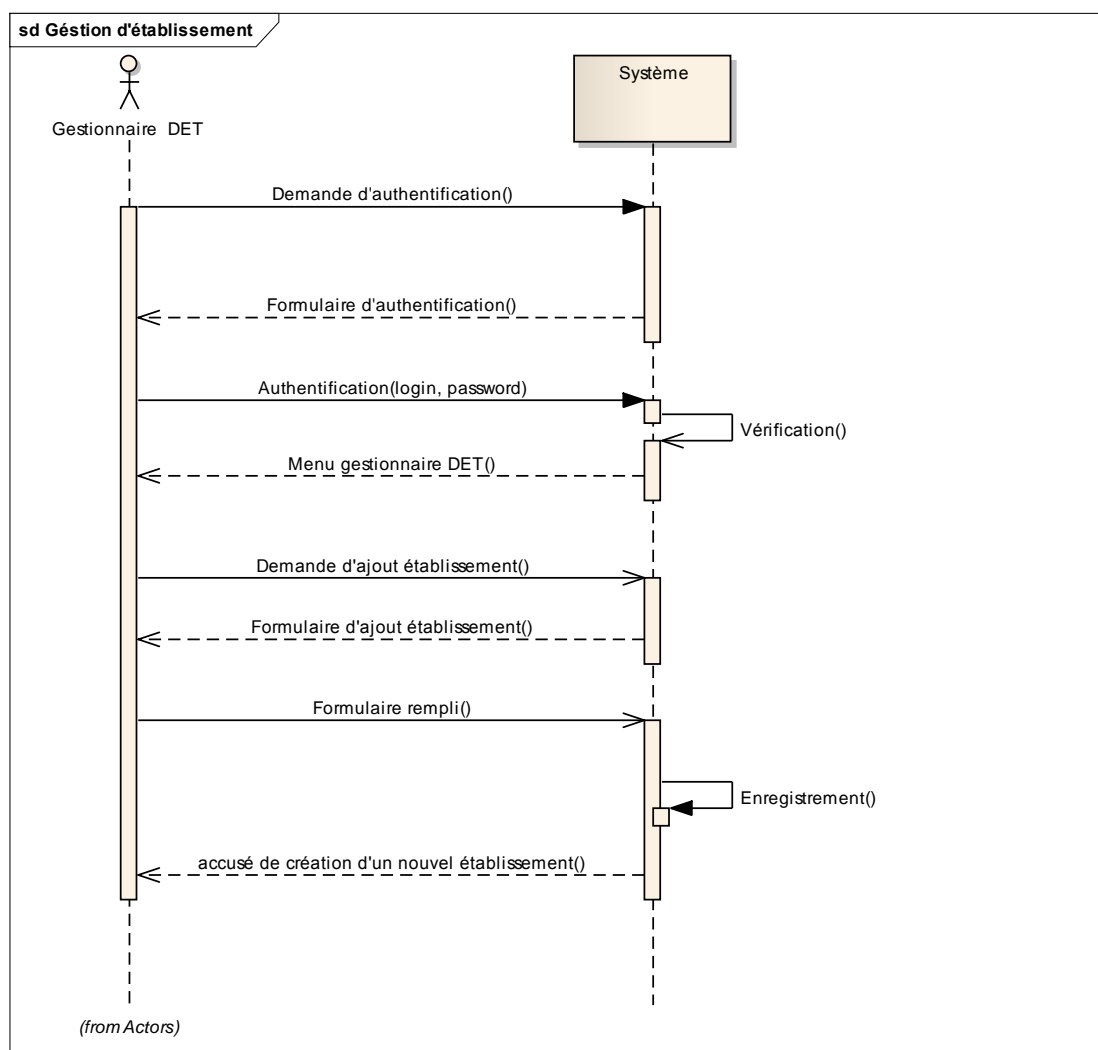


Figure II-9: Diagramme de séquence "ajouter établissement"

- Le gestionnaire DET accède à sa session en s'authentifiant via le formulaire d'authentification.
- Le gestionnaire DET demande le formulaire de création d'un établissement.
- Le système répond en lui renvoyant le formulaire.
- Le gestionnaire DET remplit le formulaire par les informations nécessaires et valide l'ajout.
- Le système vérifie la validité des informations et met à jour la base de données.

2.5.1. Diagramme de séquence d'inscription d'un étudiant

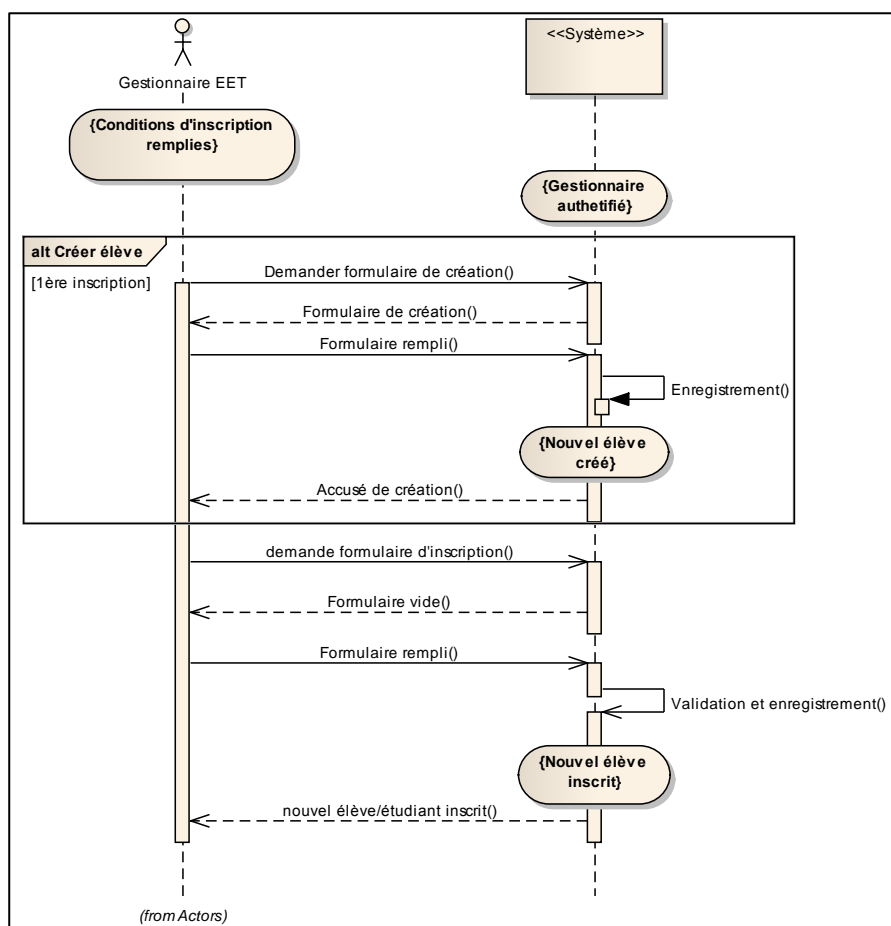


Figure II-10: Diagramme de séquence "inscrire élève"

- le gestionnaire EET accède à sa session en s'authentifiant via le formulaire d'authentification.
- Le cas d'un nouvel étudiant :

- Le gestionnaire EET demande le formulaire de création d'un étudiant
- Le système répond en lui renvoyant le formulaire de création d'étudiant.
- Le gestionnaire EET remplit le formulaire par les informations nécessaires et valide l'ajout.
- Le système vérifie la validité des informations et met à jour la base de données.
- Le gestionnaire demande le formulaire d'inscription de l'étudiant dans un établissement (affecter établissement).
- Le système répond en lui renvoyant le formulaire d'inscription.
- Le gestionnaire EET le remplit et le soumet.
- Le système valide les informations et met à jour la base de données.

2.5.2. Diagramme de séquence de l'ajout d'un employé

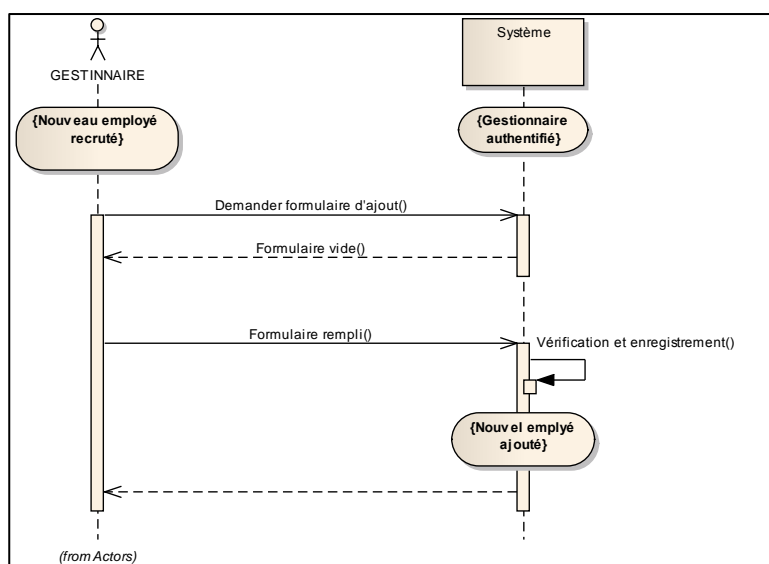


Figure II-11: Diagramme de séquence "ajouter employé"

- le gestionnaire DET accède à sa session en s'authentifiant via le formulaire d'authentification.
- Le gestionnaire DET demande le formulaire d'ajout d'un employé.
- Le système répond en lui renvoyant le formulaire d'ajout d'employé.
- Le gestionnaire EET remplit le formulaire par les informations nécessaires et valide l'ajout.
- Le système vérifie la validité des informations et met à jour la base de données.

3. Conclusion

Dans ce chapitre nous avons capturé les besoins fonctionnels de notre système et dans la phase d'analyse nous avons décrit les règles métiers en se basant sur des diagrammes des cas d'utilisation et de séquence.

Le chapitre suivant sera consacré à la conception de notre système en se basant sur les résultats obtenus dans la partie traitant l'analyse.

Chapitre 3 : Conception

Dans un premier temps nous avons réalisé les diagrammes d'interactions qui sont particulièrement utiles au concepteur pour représenter graphiquement ses décisions d'allocation de responsabilités aux classes, ensuite, et à l'aide de ces diagrammes, nous avons pu élaborer des diagrammes de classes partiels pour en arriver au diagramme de classes final.

III. CHAPITRE 3 : CONCEPTION

1. Diagramme de paquetage

La figure suivante (Figure III-1) montre les packages de notre système:

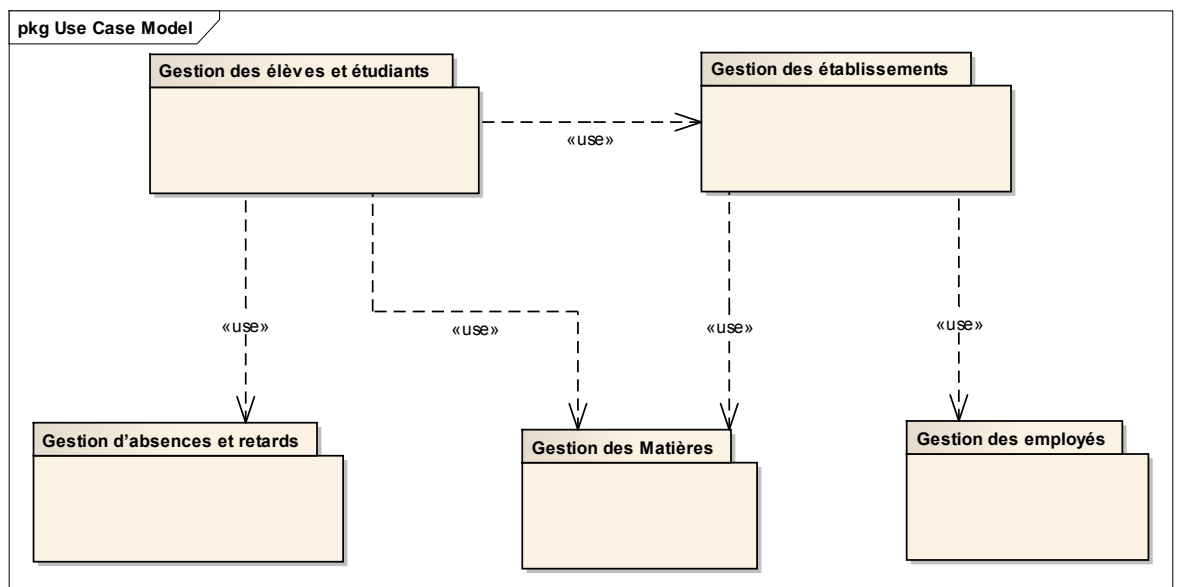


Figure III-1 Diagramme de paquetage système

▪ Gestion des établissements

- ✓ Permettre d'assurer toutes les opérations de gestion en vigueur sur une entité établissement scolaire: création, cycles autorisés ;
- ✓ Être en mesure de détenir des informations concernant l'établissement, le nom, l'adresse, la localisation, le milieu (urbain ou rural), le statut (public ou privé), la description des locaux (salles de classes, salles de bureaux, internats des élèves - filles ou garçons-, réfectoires, cuisine, magasin, bibliothèques...)

▪ Gestion des élèves et étudiants

- ✓ Assurer toutes les opérations de gestion en vigueur sur une entité élève/étudiant: première inscription, réinscription, transfert d'inscription, niveau, cycle, classe,
- ✓ Garder toutes les informations pertinentes qui contiennent des données personnelles (date et lieu de naissance, adresse, numéro de téléphone, nom et prénom en arabe et en français de l'élève et de ses parents ou tuteurs, ...), les informations concernant le parcours scolaire de l'élève (les établissements et classes dans lesquelles l'élève a étudié, notes, notes finales, la fréquentation, ...)

- **Gestion des enseignants**

- ✓ Conserver des informations détaillées sur un enseignant, y compris des renseignements personnels, matières enseignées (par année et cycle), spécialité, les degrés d'éducation.

- **Matières enseignées aux élèves et étudiants**

- ✓ Renseigner les types et intitulé des matières
- ✓ Permettre la gestion du volume horaire pour chaque matière et les coefficients et notes éliminatoires par niveau (primaire, collégial, secondaire et terminal) et par année scolaire/universitaire.

À partir des diagrammes de séquences élaborés dans la phase d'analyse, et en tenant compte des règles métiers nous avons pu identifier les diagrammes de classes concernant chaque package.

2. Le package « Gestion des élèves et étudiants »

Afin d'élaborer le diagramme de classes final de ce package nous avons dû passer par le diagramme d'interaction et aussi nous avons réalisé des diagrammes de classes partiels.

2.1. Diagramme d'interaction des informations personnelles de l'élève

Par rapport aux diagrammes de séquence système, nous allons remplacer le système vu comme une boîte noire par un ensemble d'objets en interaction. Pour cela, nous utiliserons les trois types de classes d'analyse, à savoir les dialogues, les contrôles et les entités.

Les classes qui permettent les interactions entre le site web et ses utilisateurs sont qualifiées de « dialogues ». C'est typiquement les écrans proposés à l'utilisateur : les formulaires de saisie, les résultats de recherche, etc.

Celles qui contiennent la cinématique de l'application seront appelées « contrôles ». Elles font la transition entre les dialogues et les classes métier, en permettant aux écrans de manipuler des informations détenues par un ou plusieurs objet(s) métier.

Celles qui représentent les règles métier sont qualifiées d'« entités ». Elles proviennent directement du modèle du domaine, mais sont confirmées et complétées cas d'utilisation par cas d'utilisation.

Dans notre cas et pour ce package nous présentons le diagramme d'interactions suivant:

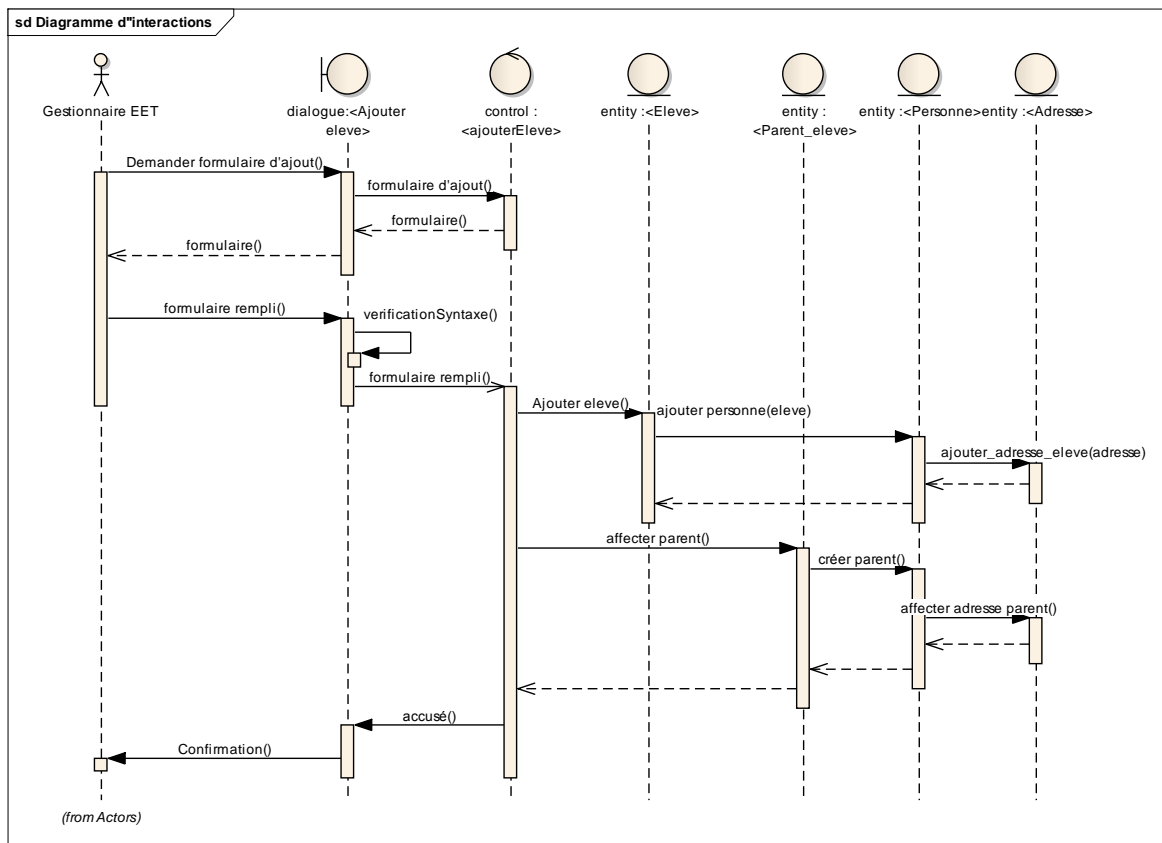


Figure III-2: Diagramme d'interaction d'ajout d'un élève

Pour ajouter un nouvel élève, on passe par une interface graphique qui représente la couche présentation. Après avoir validé les informations qui seront envoyés, nous passons par le contrôleur qui va gérer l'accès aux ressources de notre système, pour enfin arriver aux entités, et plus précisément à la couche model. D'après ce diagramme nous avons pu identifier des entités qu'on va modéliser dans le diagramme de classes partiel ci-dessous.

2.2. Diagramme de classes des informations personnelles de l'élève

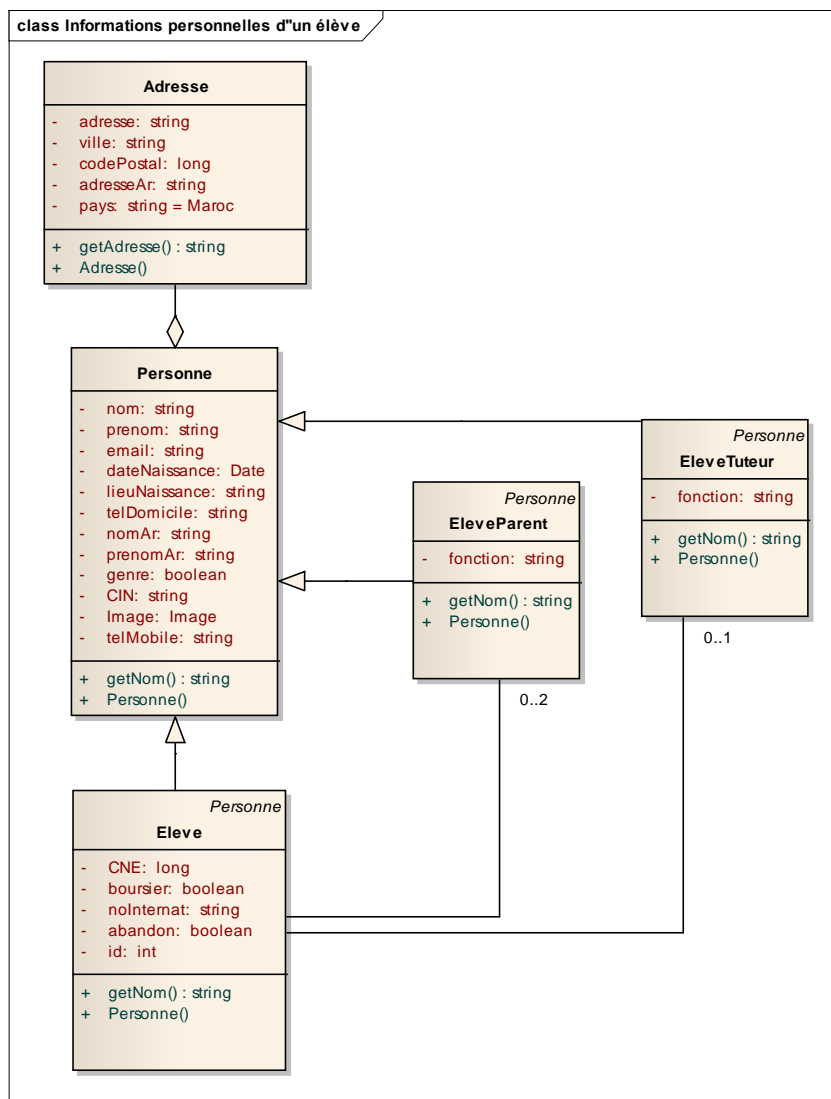


Figure III-3: Diagramme de classes des informations personnelles de l'élève

Cette figure (**Erreur ! Source du renvoi introuvable.**) montre les différentes classes concernant la gestion des informations personnelles d'un élève, on peut remarquer qu'on dispose d'une classe « *personne* » qui regroupe les informations communes aux élèves et à leurs parents et/ou leurs tuteurs. Les trois classes « *élève* », « *EleveParent* » et « *EleveTuteur* » héritent de la classe « *personne* ». Parmi les finalités de cet héritage est de gagner du temps en effectuant les opérations CRUD (*Create, Read, Update, Delete*) pendant la phase de réalisation.

L'entité « Adresse » est un objet complexe qui contient plusieurs attributs (ville, rue, numéro, pays...) qui le définissent, de ce fait, l'adresse est modéliser non pas par un simple attribut de la classe « personne » mais d'une classe qui est en relation d'agrégation avec celle-ci.

2.3. Diagramme d'interaction d'inscription

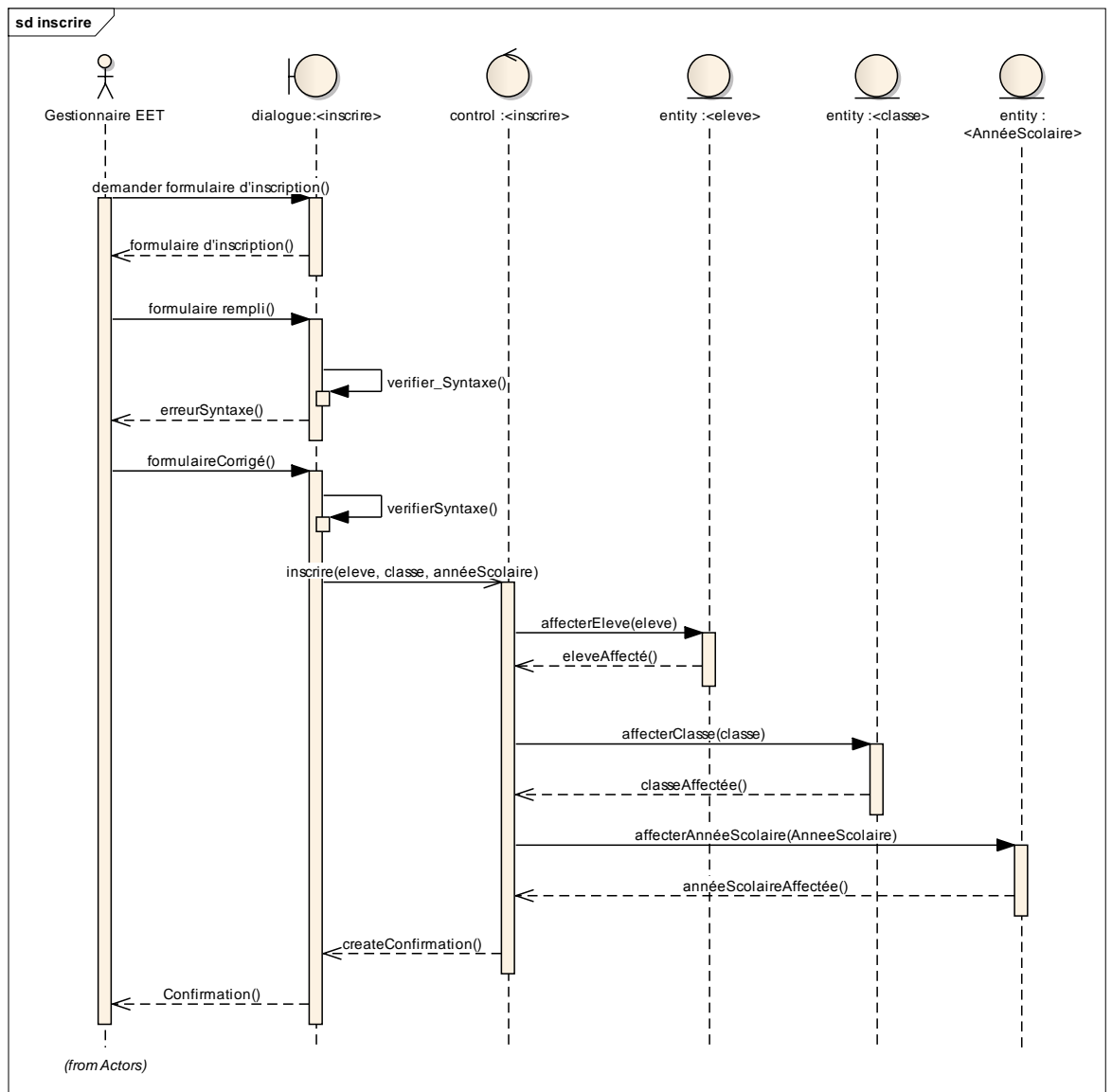


Figure III-4: Diagramme d'interaction d'inscription d'un élève

D'après ce diagramme d'interactions nous avons pu identifier les entités qui nous serons utiles pour concevoir le diagramme de classes partiel suivant.

2.4. Digramme de classes d'inscription

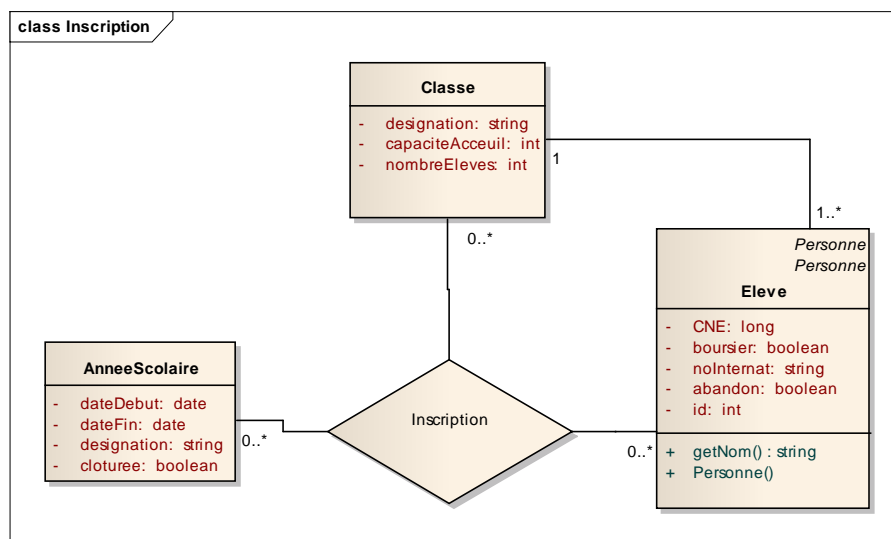


Figure III-5: Diagramme de classes d'inscription

Ce diagramme regroupe les trois classes « AnneeScolaire », « Classe », et « Eleve » contribuant à effectuer une inscription. Elles sont mises dans une relation d'association de type « n-aire » exactement « 3-aire ». Il s'agit d'une association promue au rang de classe. Elle possède tout à la fois les caractéristiques d'une association et celles d'une classe et peut donc porter des attributs qui se valorisent pour chaque lien.

En ce qui concerne les cardinalités, et par exemple l'association entre « Eleve » et « classe » est traduite ainsi : un « Eleve » n'appartient qu'à une seule classe et que « classe » possède plusieurs « Eleve » et au moins un.

2.5. Diagramme de classes de parcours scolaire

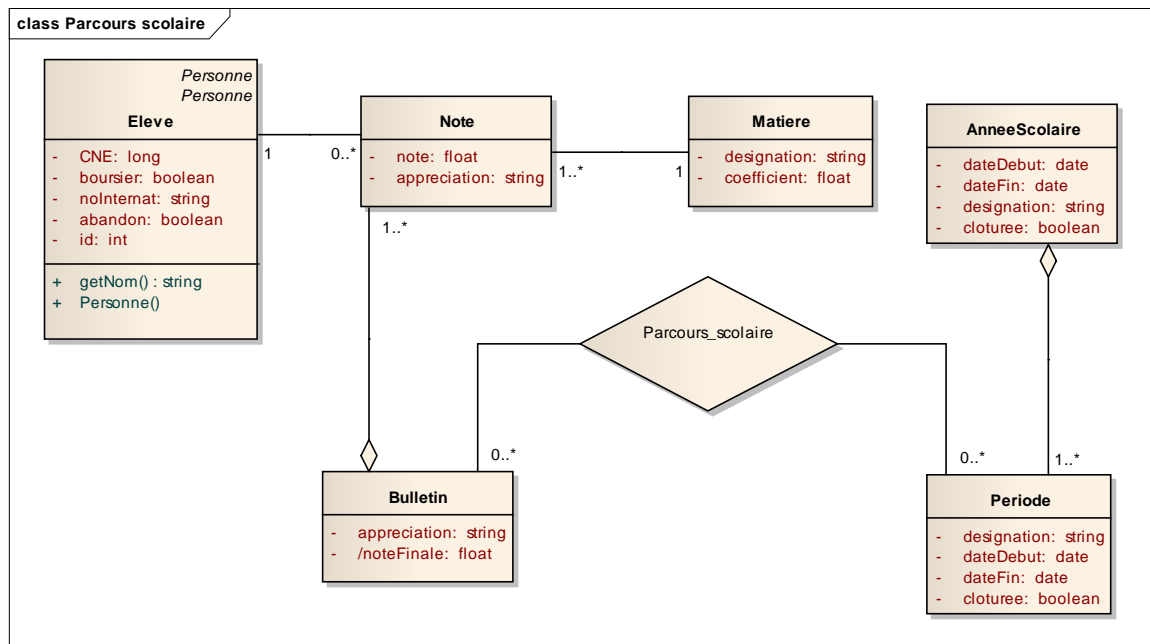


Figure III-6 : Diagramme de classes de parcours scolaire

L'association de type 2-aire entre la classe « Periode » et la classe « Bulletin » est une classe d'association qui contient le résumé du parcours scolaire d'un élève. Le parcours scolaire regroupe des informations propres à chaque élève qui sont : une note pour chaque matière. Chaque matière a un coefficient que l'on utilise avec la note pour calculer la note finale du bulletin. Chaque Bulletin est associé à un semestre de l'année scolaire

D'après les diagrammes de classes partiels ci-dessus, on a pu élaborer le diagramme de classes global du package « gestion des élèves et étudiants » :

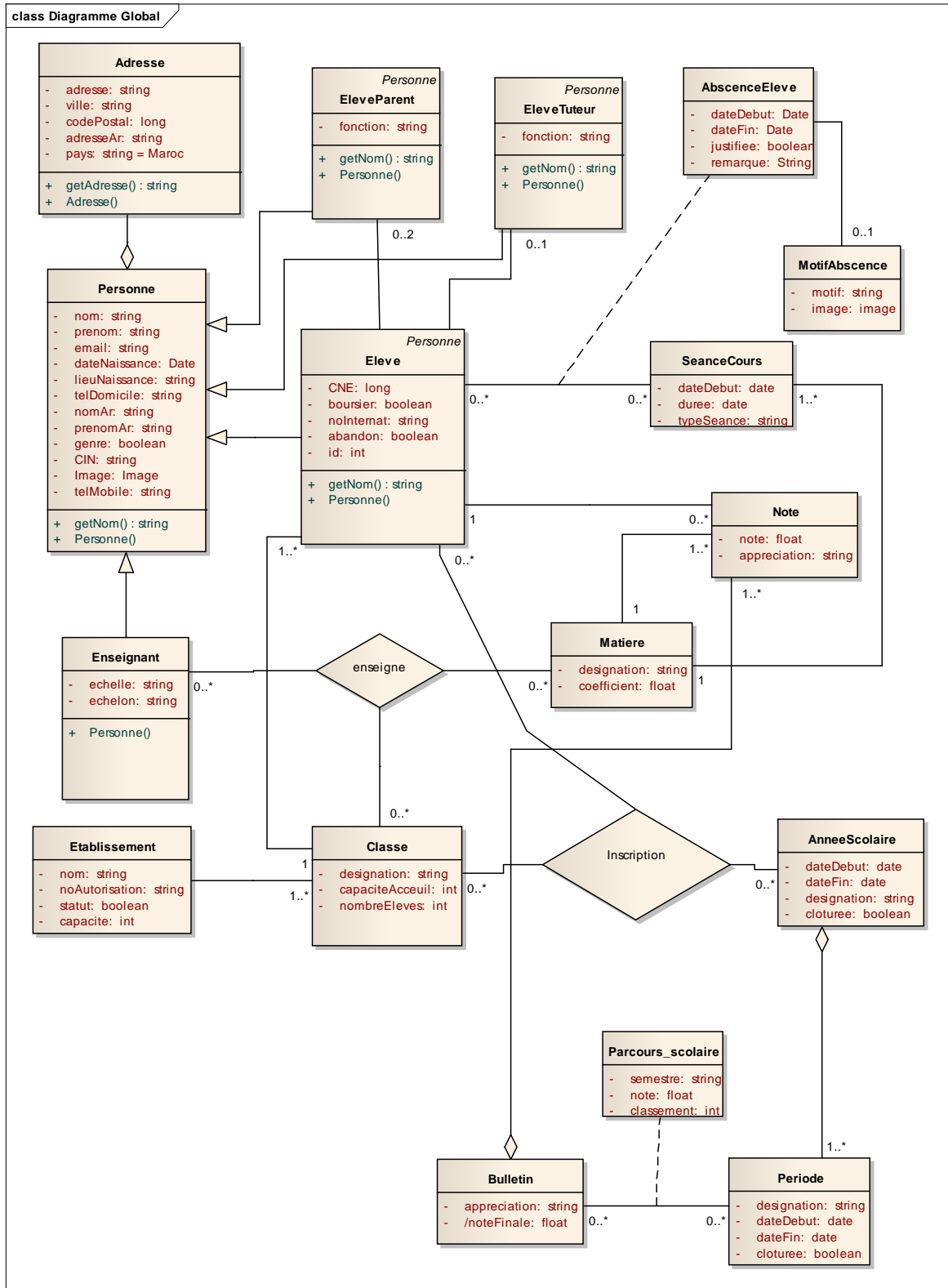


Figure III-7 : Diagramme de classes du package "gestion des élèves et étudiants"

On peut remarquer que ce diagramme global se compose en plus des diagrammes de classes partiels qu'on a élaborés ci-dessus de d'autres portions qui seront détaillés après. Cela montre bien la dépendance entre les packages.

3. Le package « Gestion des établissements »

En suivant la même démarche qu'on a suivie pour concevoir le package « gestion des élèves et étudiants », nous avons réalisé le diagramme de classes du package « gestion des établissements » à partir des diagrammes partiels suivants, et aussi à partir des dépendances existantes entre les packages de notre système.

3.1. Gestion des cycles, niveaux, et classes D'un établissement

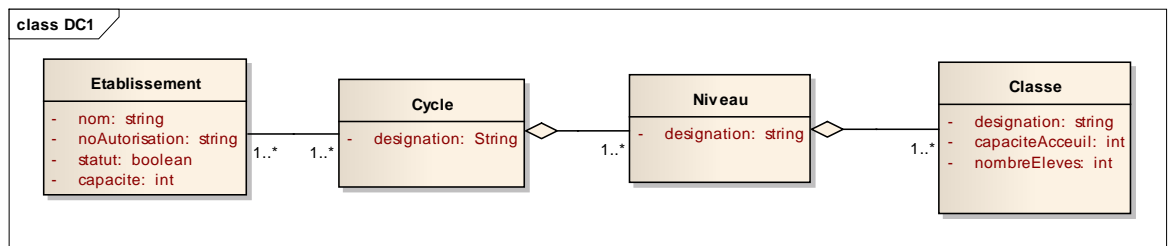


Figure III-8: Diagramme de classes de gestion des cycles, niveaux et classes

Chaque établissement propose un ou plusieurs cycles d'enseignement. Chaque cycle est composé de plusieurs niveaux, et chaque niveau contient une ou plusieurs classes.

3.2. La description des locaux

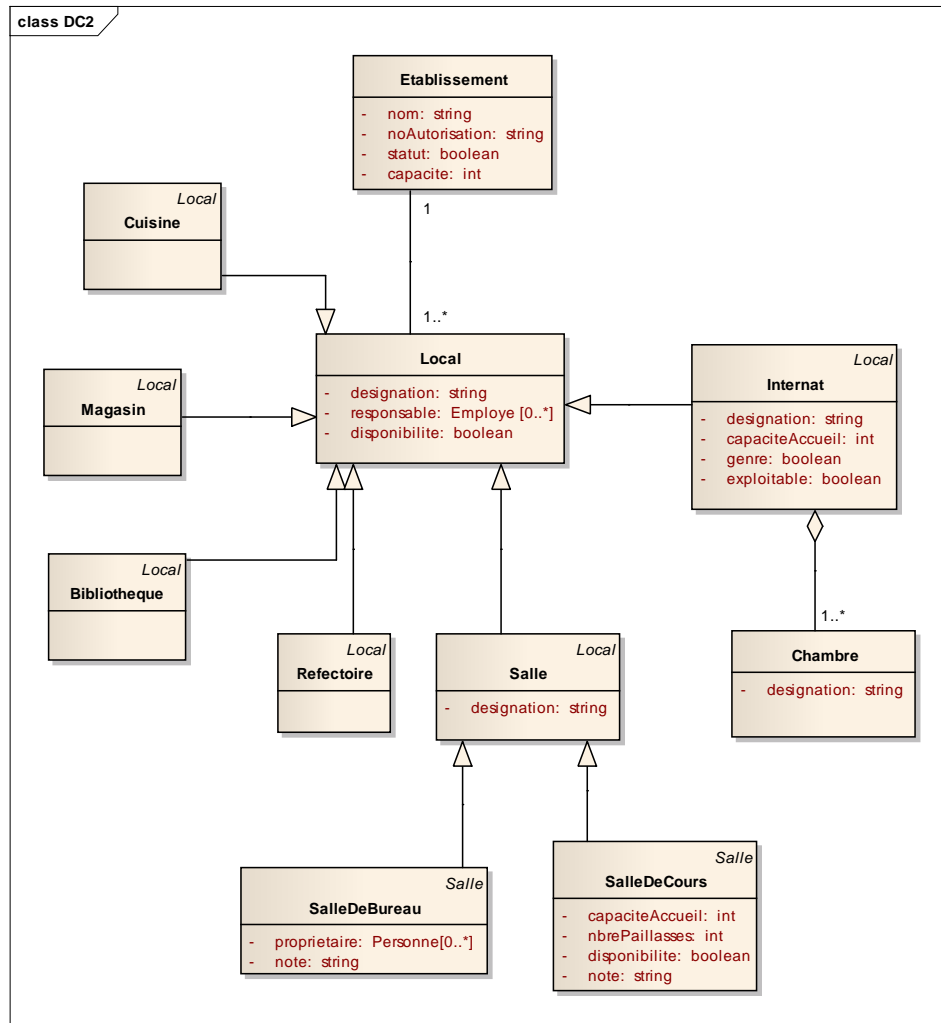


Figure III-9 : Diagramme de classes des locaux

Chaque établissement possède un ou plusieurs locaux, et chaque local a ses propres caractéristiques qui le décrivent, à titre d'exemple : un établissement a plusieurs salles qu'ils soient des salles de cours ou bien des salles de bureaux, un internat se compose de plusieurs chambres ...etc.

3.3. Définition des postes des cartes administratif, enseignants, et agents

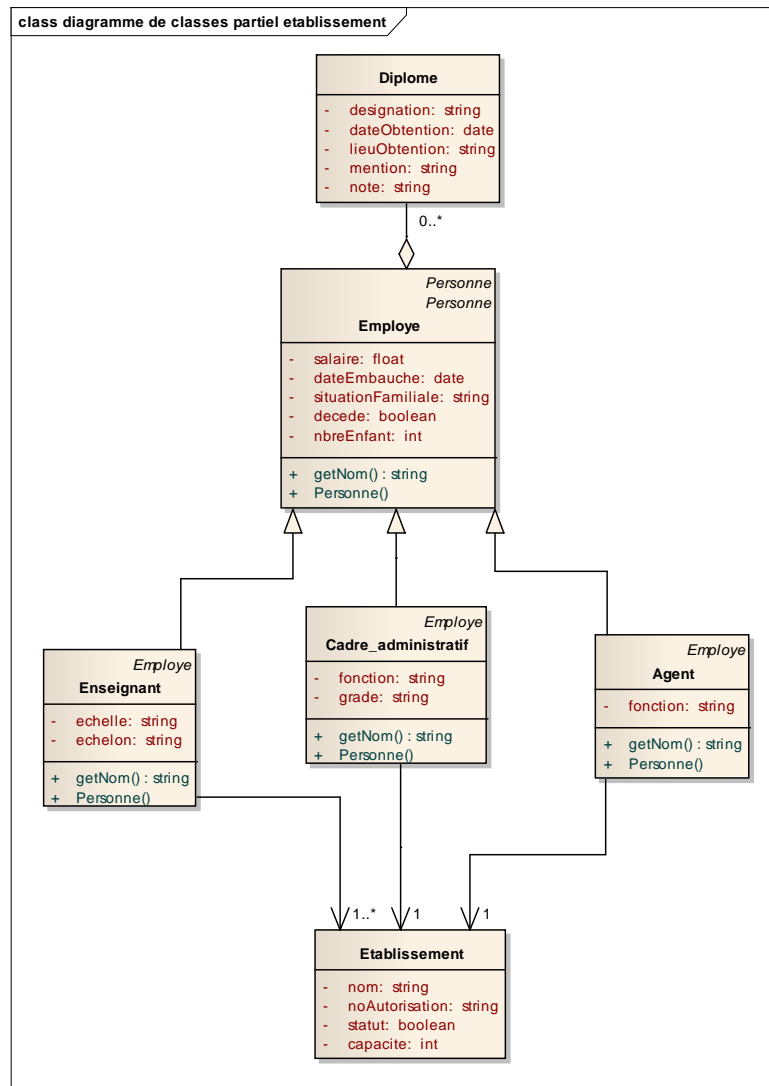


Figure III-10: Diagramme de classes des employés et du personnel

Chaque employé à 0 ou bien plusieurs diplômes, on pourra avoir un agent qui n'a aucun diplôme comme on peut trouver un enseignant qui possède plusieurs diplômes. La classe «Employe » est une classe mère qui regroupe les informations communes à tous les employés.

3.4. Diagramme Global

En regroupant les diagrammes de classes partiels ci-dessus nous avons élaboré le diagramme de classes global du package « gestion des établissements » suivants :

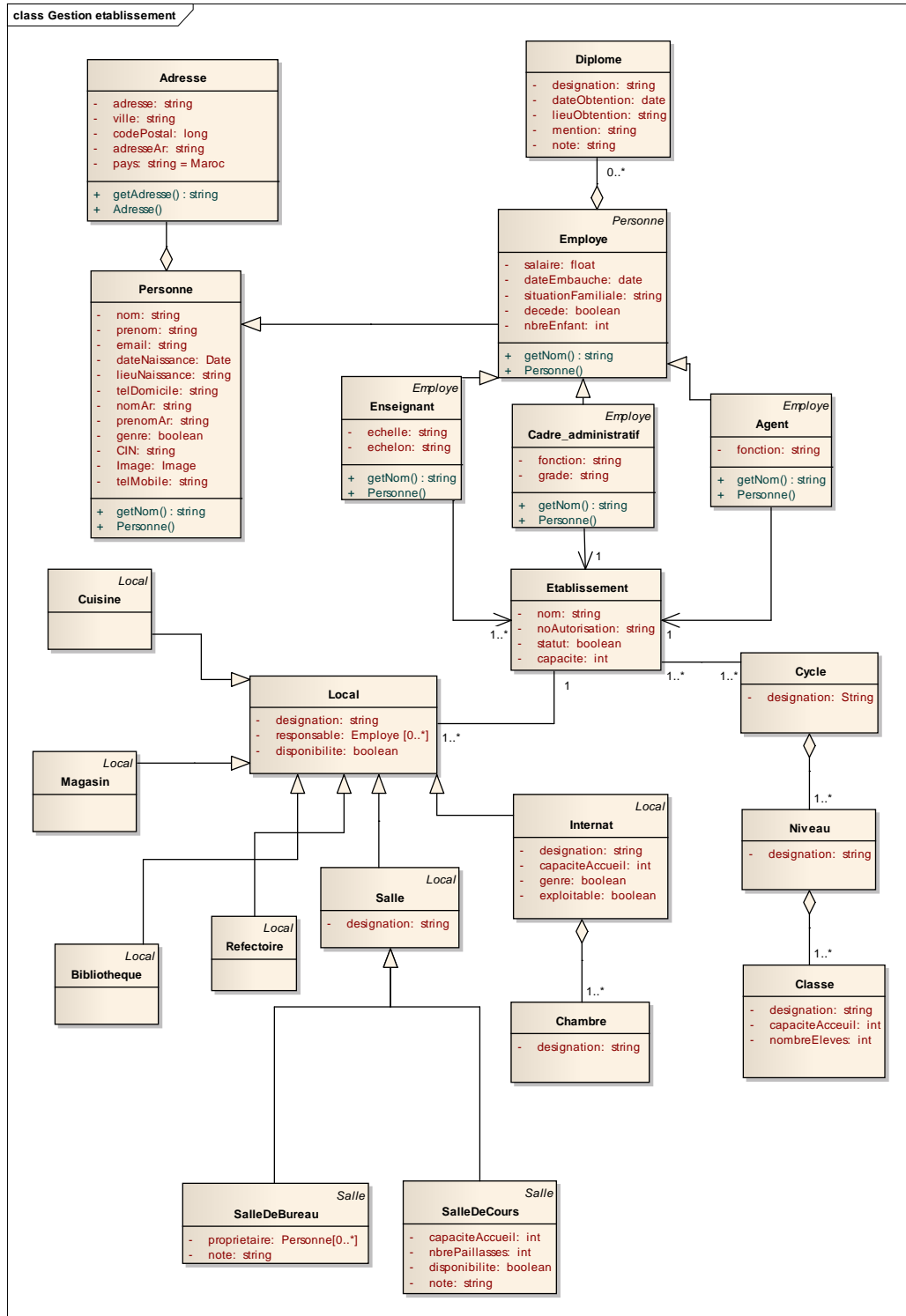


Figure III-11: Diagramme de classes du package "gestion des établissements"

4. Conclusion

Nous avons présenté dans ce chapitre une vue conceptuelle de la solution à mettre en place. Ainsi nous avons présenté les diagrammes d'interactions d'UML pour mieux comprendre les fonctionnalités offertes et qui nous étaient utiles pour élaborer le diagramme de classes global.

Dans le chapitre suivant, nous allons aborder le dernier chapitre consacré à la réalisation, dans lequel on va présenter l'environnement de travail de notre projet, ainsi que quelques interfaces représentant les fonctionnalités de notre application.

Chapitre 4 : Mise en œuvre

Dans ce chapitre, nous présentons l'étude technique et la mise en œuvre de la solution. En effet, nous allons dans un premier temps décrire le design pattern MVC et l'architecture JEE d'une part, et d'une autre part l'architecture logicielle et les outils et les Frameworks techniques utilisés pour le développement. Ensuite, on va expliquer la démarche suivie durant la phase de réalisation. Finalement, nous allons illustrer certaines fonctionnalités de l'application à travers quelques interfaces.

IV. CHAPITRE 4 : MISE EN OEUVRE

1. Le design pattern MVC

Le Modèle-Vue-Contrôleur organise l'interface Homme-machine d'une application logicielle en un modèle (objet métier, encore modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, traitement), chacun ayant un rôle précis dans l'interface. Ce design pattern impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur [MVC].

2. Java Enterprise Edition

JEE est une plate-forme fortement orientée serveur et dédiée au développement et à l'exécution d'applications distribuées. Elle permet la simplification du processus de développement [JEE]. JEE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de JEE.

L'architecture d'une application se découpe idéalement en au moins trois tiers :

- La partie cliente : il s'agit de la couche présentation correspondante à l'interface homme machine (IHM), c'est la partie qui permet le dialogue avec l'utilisateur.
- La partie métier : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans).
- La partie donnée : c'est la partie qui stocke les données dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise.

3. Outils et Frameworks utilisés

3.1. Le framework JSF 2.0

Java Server Faces est un Framework Java EE utilisé pour le développement d'applications web. Il propose une librairie de composants pour le développement de pages

web et une API de gestion de balise XHTML, d'évènement, de conversion de données, de validations de saisies, de définition de la navigation ou encore le support de l'internationalisation. JSF implémente l'architecture MVC, le modèle représenté par la couche Managed Bean associé à une entité de persistance, la vue est représentée par les pages XHTML et le contrôleur par la Servlet *Faces Servlet* livrée avec JSF.

3.2. Enterprise Java Beans (ejb 3)

Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développement JEE.

Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés au sein d'un serveur applicatif (conteneur EJB) permettant de représenter des données (EJB dit entité), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit session), ou encore d'accomplir des tâches de manière asynchrone (EJB dit message). Tous les EJB peuvent évoluer dans un contexte transactionnel.

3.3. JPA 2.0

Java Persistence API (JPA) est un standard Java utilisé pour la persistance des données qui utilise le principe de mapping objet/relationnel afin de permettre de stocker les objets dans la base de données, et inversement, de pouvoir lire les données relationnelles et les transformer en objets. L'API JPA 2.0 offre les services suivants :

- ✓ La gestion de persistance,
- ✓ Un langage de requêtage évolué : Java Persistence Query Language (JPQL),
- ✓ Un mécanisme de mapping objet/relationnel ORM à partir de métadonnées (fichiers XML ou annotations),
- ✓ Une API à typage fort pour le requêtage.

3.4. Le Framework EclipseLink

EclipseLink est un Framework open source de mapping objet-relationnel (ORM) pour les développeurs Java. Il fournit une plateforme puissante et flexible permettant de stocker des objets Java dans une base de données relationnelle et/ou de les convertir en documents XML. Il supporte un certain nombre d'API relatives à la persistance des données et notamment la JPA.

3.5. Oracle 10g

Oracle est un SGBD (système de gestion de bases de données) édité par la société du même nom (Oracle Corporation), leader mondial des bases de données. Oracle permet d'assurer la manipulation, la confidentialité, l'intégrité et la cohérence des données. Il gère aussi les accès concurrents ainsi que la sauvegarde et la restauration des données.

4. Architecture technique

L'architecture technique présente les différents outils utilisés dans le développement de chaque partie du projet. La figure suivante présente l'architecture technique de notre application :

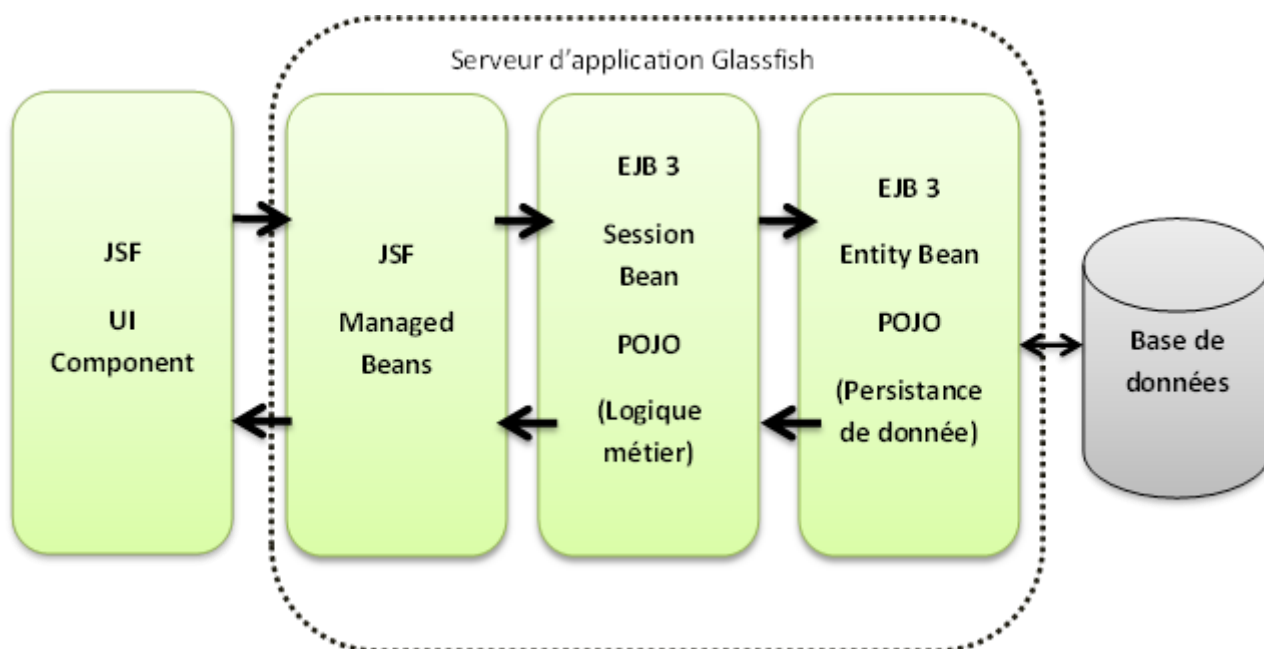


Figure IV-1: L'architecture technique de l'application

La couche présentation est constituée d'un ensemble de composants JSF. Les JSF Managed Beans sont utilisés pour faire le lien entre la couche présentation et la couche métier. Ils permettent ainsi de récupérer les données saisies par les utilisateurs, les valider et les transmettre à la couche métier.

La couche métier est un ensemble de composants EJB session qui encapsulent la logique métier de l'application. Les EJB entités représentent le modèle des données. Ceux-ci sont directement persistés dans la base de données grâce à l'api de persistance JPA.

5. Démarche de développement

Afin de mieux organiser la phase de mise en œuvre, nous avons veillé à procéder par étapes. Ainsi, nous avons procédé de la manière suivante pendant la phase de développement:

- **Créations de classes entités à partir de la base de données :**

Dans ce projet nous allons utiliser une base de données. On ne va pas manipuler des données directement en SQL mais sous forme d'objets particuliers, objets persistants qu'on appelle des "classes entités".

Généralement, une classe entité correspond à une table, elle porte le même nom (aux majuscules/minuscules près), et ses attributs correspondent aux colonnes de la table. Les instances de cette classe entité seront des objets correspondant à des lignes dans cette table.

- **Création des Stateless Session Bean pour la gestion des classes entités :**

La gestion des différentes classes entités sera centralisée dans les *Stateless Session Bean*. On va donc créer au niveau de chaque *Stateless Session Bean* une "façade" pour les opérations élémentaires sur les entités (création, suppression, recherche, modification), ainsi que les différentes méthodes métiers agissant sur les classes entités.

- **Création des pages JSF et les Managed Bean associés :**

Dans cette étape nous avons développé les interfaces graphiques suivant le modèle qu'on a établi dans la partie conception. Cette étape concerne l'interaction entre le système et les requêtes de l'utilisateur à travers les interfaces Homme-Machine que nous avons développé en utilisant les pages JSF (Facelet), à chacune de ces interfaces est associée une classe Managed Bean.

Une page JSF contient des balises qui décrivent les composants qui représenteront la page sur le serveur. Elle contient aussi des parties en EL, et utilise souvent une (ou plusieurs) bibliothèque de composants, mais ne contient pas de traitements (pas de code

Java ou autre code comme dans les pages JSP pour créer les pages affichées à l'utilisateur).

Un Managed Bean (dit aussi Backing Bean) assure les différents traitements liés directement à l'interface utilisateur, et fait appels à des EJB ou des classes Java ordinaires pour effectuer les traitements qui ne sont pas liés directement à l'interface utilisateur.

Les Managed Beans associés aux pages JSF, font donc la liaison entre les composants JSF de l'interface, les valeurs affichées ou saisies dans les pages JSF et le code Java qui effectue les traitements métier.

6. Interfaces de l'application

Durant cette section, nous présentons un scénario d'utilisation de l'application par quelques interfaces.

6.1. Authentification

Pour pouvoir accéder au menu principal de l'application, il est impératif de s'authentifier, et cela en renseignant un identifiant et un mot de passe.

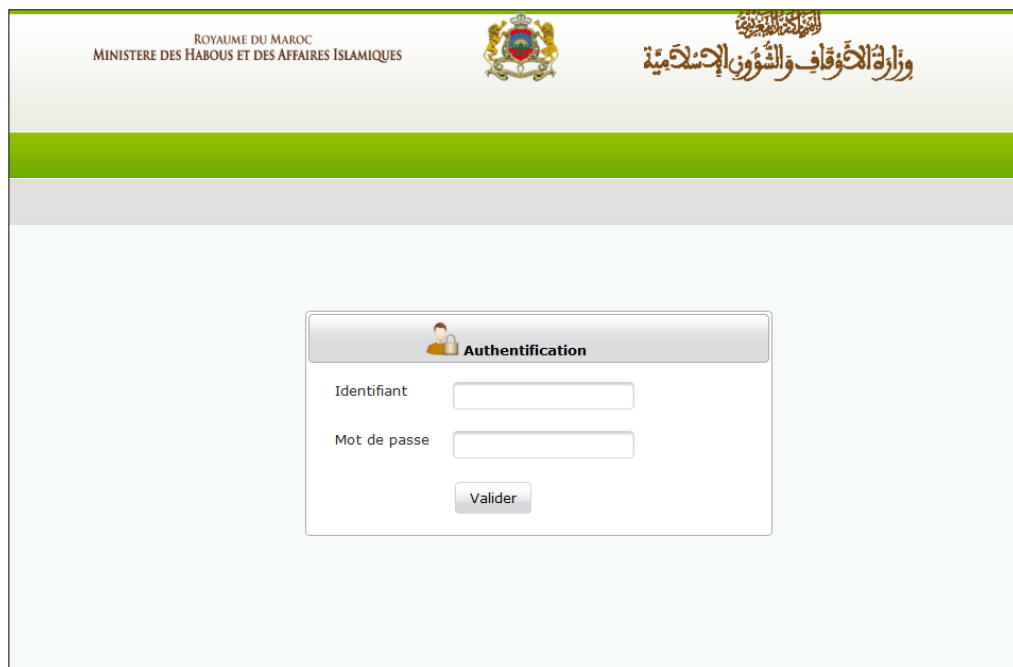


Figure IV-2: Interface de l'application: authentification

Lorsque l'utilisateur se connecte, celui-ci voit s'afficher un menu, selon le profil qu'il possède. L'utilisateur connecté est redirigé vers l'écran contenant le menu suivant :



Figure IV-3: Interface de l'application: Menu principal

On voit bien dans ce menu les différents modules de gestion que nous avons développés. L'utilisateur pour accéder à un module n'a qu'à cliquer sur l'icône correspondante.

Dans ce qui suit on va présenter les différentes interfaces représentant les différentes fonctionnalités du module de gestion des étudiants.

6.2. Gestion des étudiants

▪ Consultation de la liste des étudiants :

En accédant au module gestion des étudiants, l'utilisateur est redirigé vers une page listant les différents étudiants enregistrés sous forme d'un tableau, avec la possibilité de consulter, modifier ou supprimer les informations concernant un étudiant, et cela en cliquant sur l'icône correspondante à chacune de ses action.

Pour ajouter un nouvel étudiant, l'utilisateur n'a qu'à cliquer sur le bouton se trouvant au-dessous du tableau.

The screenshot shows the 'LISTE DES ÉLÈVES' interface. At the top, there is a header with the text 'ROYAUME DU MAROC' and 'MINISTÈRE DES HABOUS ET DES AFFAIRES ISLAMIQUES' on the left, the national emblem in the center, and the Arabic name of the ministry on the right. Below the header is a green navigation bar. On the left, there is a sidebar menu with items: Menu, Absences, Années scolaires, Les classes, Cours, Délégation, Elève, Enseignants, Etablissements, and Inscriptions. The main content area is titled 'LISTE DES ÉLÈVES' and includes a 'Se Déconnecter' link. Below the title, there is a table with 10 columns: Nom, Prenom, Genre, Cne, Cin, DateNaissance, LieuNaissance, TelephoneMobile, Email, and three action icons (search, edit, delete). The table contains three rows of student data. Below the table is a button labeled 'Ajouter un nouvel élève' and a link for 'Menu principal'.

Nom	Prenom	Genre	Cne	Cin	DateNaissance	LieuNaissance	TelephoneMobile	Email			
Ahmadi	ahmed	M	234569	JB380234	05/10/2012	rabat	0666678988	a.ahmadi@gmail.com			
Belhaj	Mohamed	M	123211	AZ123344	05/16/2012	azilal	06554422	m.belhaj@yahoo.fr			
BENOMAR	Noureddine	M	KJ456665	12332455	01/06/2008	Tanger	06988383838	n.benomar@hotmail.fr			

Figure IV-4: Interface de l'application: liste des élèves ajoutés

• Ajout d'un nouvel étudiant :

En cliquant sur le bouton « Ajouter un nouvel élève », une page contenant un formulaire constitué de quatre onglets s'affiche. Chaque onglet contient un ensemble d'informations nécessaires pour enregistrer un élève.

Le premier onglet « Informations personnelles » contient le nom, prénom, genre, code national élève, CIN ainsi que la date et le lieu de naissance.

Le deuxième onglet « Contact » contient l'adresse, le téléphone et l'email de l'élève. Le quatrième onglet quant à lui contient des informations sur le parent de l'élève, alors que le dernier onglet est utilisé pour confirmer les informations saisies dans les onglets précédents.

Figure IV-5: Interface de l'application: formulaire d'ajout d'un élève

▪ **Consultation des informations d'un élève :**

En cliquant sur l'icône « loupe », une page contenant les détails d'un élève sous forme de tableau s'affiche. Dans cette même page l'utilisateur a la possibilité de supprimer ou de modifier les informations concernant l'élève.

Informations de l'élève	
Nom:	Ahmadi
Prenom:	ahmed
Genre:	M
Cne:	234569
Cin:	JB380234
DateNaissance:	05/10/2012
LieuNaissance:	rabat
Adresse:	rue atlas N34 agdal
TelephoneFixe:	05334434
TelephoneMobile:	0666678988
Email:	a.ahmadi@gmail.com
Nom du père	Alaoui said

Figure IV-6: Interface de l'application: afficher les informations un élève

- **Modifier les informations d'un élève :**

En cliquant sur l'icône « stylo », un écran semblable à celui de la consultation s'affiche, mais cette fois-ci, les différentes informations de l'élève sont éditables.

Informations de l'élève	
Nom:	A
Prenom:	ahmed
Genre:	M
Cne:	234569
Cin:	JB380234
DateNaissance:	05/10/2012
LieuNaissance:	rabat
Adresse:	rue atlas N34 agdal
TelephoneFixe:	05334434
TelephoneMobile:	0666678988
Email:	a.ahmadi@gmail.com
Idparent:	Alaoui

Figure IV-7: Interface de l'application: Modifier un élève

7. Conclusion

Ce chapitre avait pour but de décrire l'environnement technique et les différentes étapes de la mise en œuvre de l'application.

En effet, nous avons décrit dans un premier temps l'architecture technique. Ensuite, nous avons recensé les différentes étapes de la phase de développement, et enfin nous avons présenté quelques interfaces décrivant certaines fonctionnalités de l'application.

Conclusion générale

Notre projet de fin d'études que nous avons effectué au sein de VisualSoft, consistait en l'étude, l'analyse, la conception et la réalisation des modules de gestion des établissements, des élèves, et des enseignants. Ces modules font partie du projet lancé par le ministère des Habous et des affaires islamiques pour la mise en œuvre d'une solution intégrée de gestion de la scolarité dans les niveaux préscolaire, primaire, collégial, secondaire et terminal de l'enseignement traditionnel marocain.

Ce projet s'est déroulé selon quatre phases. Dans la première phase, nous étions amenés à faire une étude fonctionnelle dont le but est de spécifier et d'analyser les besoins. Dans la deuxième phase nous nous sommes penchés sur la conception de notre système afin de bien structurer la couche métier de l'application. La troisième phase a été consacrée à l'étude technique où l'on a détaillé l'architecture technique et les outils et les Frameworks utilisés dans le développement. Finalement, la quatrième phase a été consacrée à la réalisation et à la mise en œuvre des différents modules de notre système.

L'approche de conduite de projet RUP qu'on a mis en œuvre prend en compte les changements continuels qui peuvent survenir lors de la réalisation, et cela pour que le résultat obtenu soit en accord avec les besoins réels des utilisateurs. Cette approche combinée avec l'utilisation des Frameworks de développement, permet de réduire considérablement le coût réalisation et de la maintenance applicative, et aussi d'avoir une architecture de l'application qui se veut flexible et extensible, ce qui facilitera son intégration avec les autres modules futurs à développer.

Ce stage a été pour nous l'occasion de faire le lien entre nos connaissances académiques et le monde professionnel. En effet, il nous a permis de développer nos compétences techniques, d'approfondir nos connaissances théoriques et les mettre en pratique. Cette expérience a aiguisé nos capacités d'analyse et de synthèse, et nous a permis de renforcer nos connaissances concernant l'architecture JEE, notamment le Framework JSF 2, les EJB ainsi que l'API JPA.

Enfin, ce stage fut une expérience très enrichissante pour nous sur les deux plans personnels et professionnels. En effet, il a été l'occasion de découvrir le dynamisme et l'enthousiasme qui caractérisent l'équipe de VisualSoft. Les réunions régulières effectuées avec les encadrants à VisualSoft nous ont permis de mettre en œuvre les concepts de gestion de projets acquis à l'EMI.

Bibliographie

- Jérôme LAFOSSE, « *Développement n-tiers avec Java EE* », édition ENI, collection Epsilon (7 mars 2011).
- Pascal Roques, « *UML2 : Modéliser une application web* », édition EYROLLES, collection les Cahiers du Programmeur (4^e édition : 02/10/2008).
- Antonio Goncalves, « *Java EE 5 : EJB 3, JPA, JSP, JSF, GlassFish* », édition EYROLLES, collection les Cahiers du Programmeur (2^e édition : 25/05/2007).

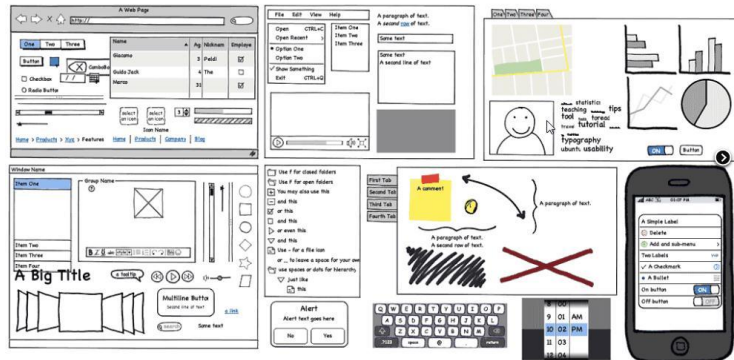
Webographie

[MVC]	<i>Model View Controller</i> , disponible sur : https://sites.google.com/site/floriangrisoni/informatique/design-pattern-mvc
[JEE]	<i>Java Enterprise Edition</i> , disponible sur : http://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm#j2ee-javaee-1
[NetBeans]	<i>Getting Started with Java EE 6 Applications</i> , disponible sur: http://netbeans.org/kb/docs/javaee/javaee-gettingstarted.html
[JSF 2]	<i>JSF 2 Templating with Facelets example</i> , disponible sur: http://www.mkyong.com/jsf2/jsf-2-templating-with-facelets-example/
[EJB 3]	<i>EJB3 Introduction</i> , disponible sur : http://www.datadisk.co.uk/html_docs/ejb/ejb3_introduction.htm
Tutoriel : EJB3.1/JPA/J SF2	<i>TP1 2011 EJB 3.1/JPA/JSF2</i> , MIAGE Nice, disponible sur : http://miageprojet2.unice.fr/Intranet_de_Michel_Buffa/Cours_composants_distribu%C3%A9s_pour_l'entreprise_%2f%2f_EJB_2009/TP1_2011_EJB_3.1%2f%2fJPA%2f%2fJSF2
[EclipseLink]	Tutorial : <i>the basic steps of developing, packaging, and deploying a Web application using the EclipseLink</i> , disponible sur: http://wiki.eclipse.org/EclipseLink/Examples/JPA/JSFTutorial

Annexe A : Outils utilisés

Balsamiq, Mockups Version 2.1.15

Proposée par les studios Balsamiq, Mockups est une application permettant de concevoir des modèles d'interface. Entre l'outil de dessin vectoriel et Microsoft Visio, celle-ci donne la possibilité à l'utilisateur de travailler de



manière collaborative à l'élaboration de croquis dans lesquels il pourra placer des croquis, des notes, des couleurs, des icônes, des fenêtres flottantes ainsi que divers outils.

Enterprise Architect Version 7.5



Enterprise Architect est un outil d'analyse et de création UML, couvrant le développement de logiciels du rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien.

Cet outil graphique basé sur Windows, peut être utilisé par plusieurs personnes et conçu pour vous aider à construire des logiciels faciles à mettre à jour. Il comprend un outil de production de documentation souple et de haute qualité.

NetBeans IDE Version 7.1.2

NetBeans IDE est un environnement de développement qui s'adapte aux langages de programmation (Javascript, Python, PHP, Groovy, C/C++, etc.), aux outils et aux ressources dont vous disposez. Le programme détecte automatiquement la présence de Java,



JDK, SOA, Ruby, MySQL, etc. sur votre système, ainsi que les serveurs Apache ou GlassFish, pour vous fournir les plugins nécessaires. Vous pourrez ainsi créer facilement des applications Web, des portails d'entreprise, des logiciels multiplateforme sous Java, des logiciels pour mobiles, etc. Vous disposerez de nombreux outils d'édition, d'un debugger, d'une module de prévisualisation, de modèles et de bibliothèques, de fonctions de complétion et d'implémentation, etc. La dernière version de NetBeans IDE supporte dorénavant les spécifications du langage de développement Java SE 7. Il s'intègre également avec les serveurs Oracle WebLogic et supporte Oracle Database et GlassFish 3.1. On note au passage la facilité d'édition HTML5 et un nouveau GridBagLayout conçu pour améliorer le développement d'interfaces utilisateur et l'édition de code Java.

Adobe Dreamweaver CS5



Adobe Dreamweaver CS5 propose aux développeurs Web une plateforme de développement pour la création de sites et d'applications Web. Le logiciel propose d'utiliser des outils d'inspection CSS pour la conception et de développer et améliorer des systèmes de gestion de contenus tels que Joomla, Wordpress ou encore Drupal. Adobe Dreamweaver CS5 dispose également d'un module de tests de compatibilité avec les navigateurs grâce à l'intégration avec Adobe BrowserLab.

Annexe B : Le cycle de vie à phases multiples d'une page JSF

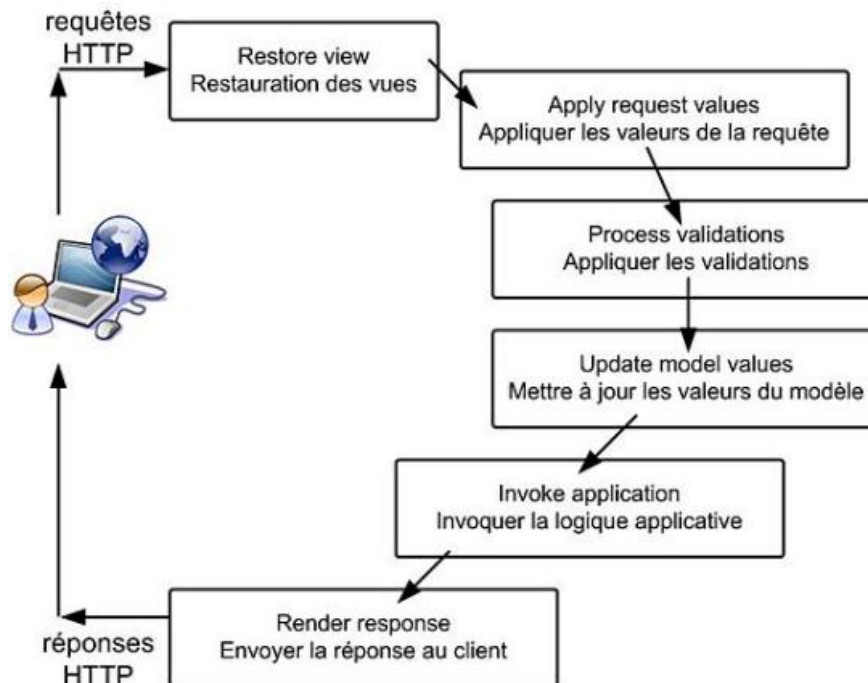
Après une présentation rapide de la notion de cycle de vie à phases multiples d'une page JSF, cette annexe décrit chacune des phases en question. Elle apporte des informations purement théoriques, expliquant comment une application JSF assure la prise en charge de la gestion des événements ainsi que la conversion et la validation des données.

Principe :

Le cycle de vie d'une page JSF est partitionné en plusieurs phases, dans le but de pouvoir répondre efficacement aux attentes liées à cette technologie: capture d'événements au niveau des composants graphiques, validation ou conversion de composants, liaison entre composants d'interface et objets s'exécutant côté serveur. Deux types de requêtes doivent être distingués :

- la requête initiale au cours de laquelle l'internaute accède pour la première fois à la page JSF,
- la requête dite postback correspondant le plus souvent à la validation d'un formulaire précédemment chargé dans le navigateur du client.

Présentation des différentes phases :



a) Phase de restitution de la vue (Restore view) :

Dès qu'un internaute tente d'atteindre une page JSF, la phase de restitution de vue est exécutée. Son but est d'associer une vue à la page visitée. L'application web crée une vue vierge dans le cas d'une requête initiale; elle restitue la vue précédemment associée à la page dans le cas d'une requête postback. Tous les gestionnaires d'événements ainsi que les validateurs requis par les composants graphiques de la page sont liés à la vue. Un arbre de composants est constitué. La vue est ensuite sauvegardée dans l'objet prédéfini FacesContext.

b) Phase Application des paramètres de requête (Apply request values)

À l'issue de la phase précédente, une vue est associée à la page JSF visitée. Cette vue dispose obligatoirement d'un arbre de composants. Ces composants sont alors parcourus successivement pour invoquer leur méthode decode(). Celle-ci est chargée d'analyser la requête pour attribuer au composant la valeur qui doit lui être affectée.

À la fin de cette phase, tous les composants associés à la vue disposent d'une nouvelle valeur. Les messages et les événements éventuels sont en attente de traitement par l'objet FacesContext.

c) Phase Processus de validation (Process Validation)

Dans cette étape, l'application JSF effectue toutes les validations attendues pour chacun des composants de l'arbre associé à la vue.

Pour cela, les attributs de composant définissant les règles de validation sont examinées, puis comparés à la valeur du composant. Lorsque cette valeur est effectivement incorrecte, un message d'erreur est ajouté à l'objet FacesContext et le cycle de vie se poursuit directement par la phase de traduction de la réponse: la page web est alors restituée à l'internaute, accompagnée du message d'erreur issu du processus de validation.

d) Phase Mise à jour du modèle (Update Model Values)

Chaque composant d'interface JSF peut être associé à un JavaBean côté serveur. Après s'être assurée de la validité des données saisies lors de la phase Processus de validation, l'application JSF parcourt à nouveau l'arbre de composants pour affecter la valeur de chaque composant graphique au JavaBean qui lui est associé. Là encore, si un problème de

conversion survient au moment de la mise à jour de la propriété du JavaBean, le cycle de vie de la page se poursuit directement par la phase de traduction de la réponse et un message d'erreur adapté est présenté à l'internaute.

e) Phase Appel de l'application (Invoke application)

Lorsque l'internaute valide un formulaire, ou clique sur un lien hypertexte, l'application JSF génère respectivement un objet de type "événement de formulaire" ou un objet de type "événement de commande". Ces objets sont qualifiés d'événements de niveau application : ils sont pris en charge par des gestionnaires spécifiques au cours de cette phase dont le rôle est de mentionner une URL vers laquelle la navigation est dirigée.

f) Phase de restitution de la réponse (Render response)

Cette phase correspond au moment où l'implémentation Java Server Faces rend la main au conteneur en charge des pages JSP de l'application web. Tous les composants sont alors présentés dans l'interface utilisateur, dans l'état qui est le leur au moment où survient la phase Render response

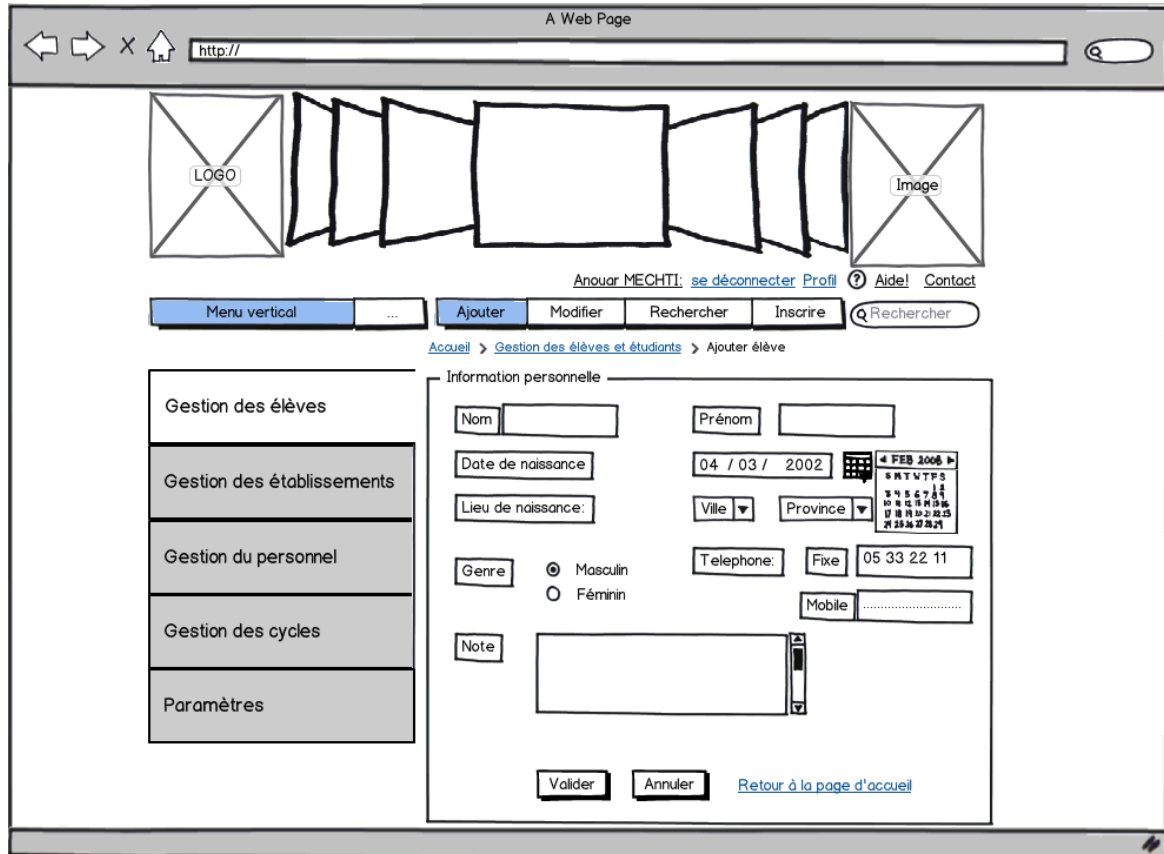
Annexe C : Les maquettes

Une maquette est un produit jetable donnant aux utilisateurs une vue concrète mais non définitive de la future interface de l'application. Cela peut consister en un ensemble de dessins réalisés avec des outils spécialisés tels que Dreamweaver, Adobe Illustrator ou plus simplement avec Powerpoint ou même Word. Par la suite, la maquette intégrera des fonctionnalités de navigation pour que l'utilisateur puisse tester l'enchaînement des écrans, même si les fonctionnalités restent fictives.

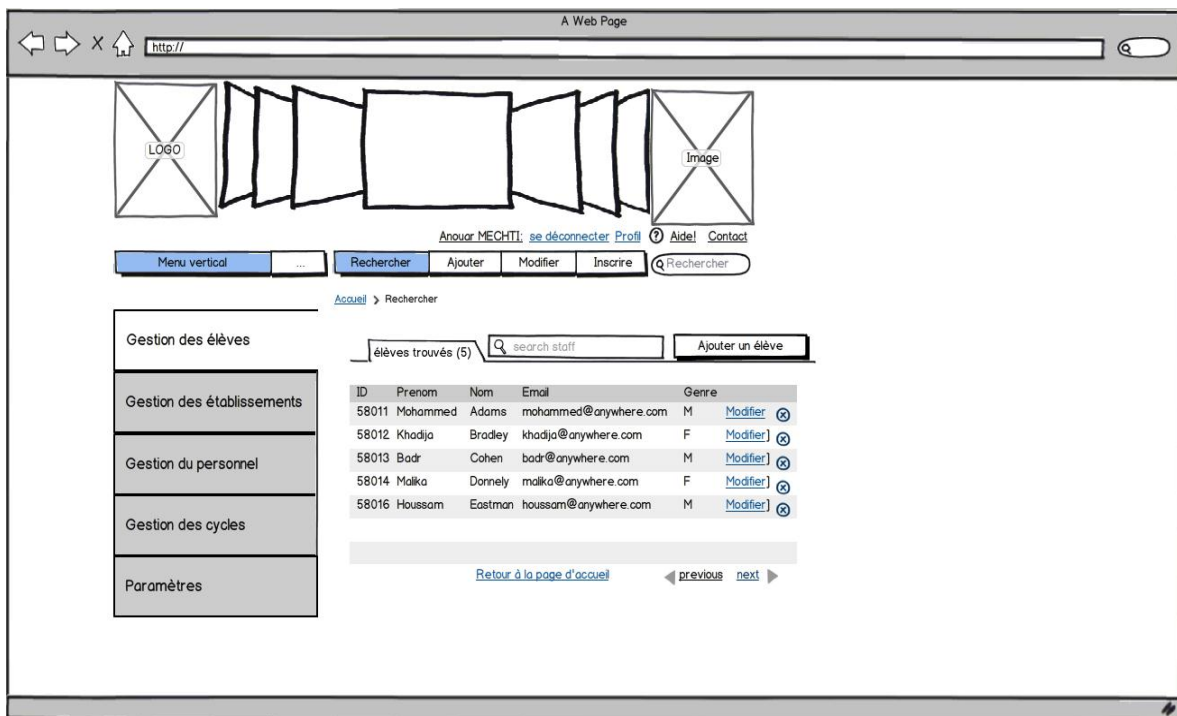
La maquette est développée rapidement afin de provoquer des retours de la part des utilisateurs. Elle permet ainsi d'améliorer la relation développeur-client. La plupart du temps, la maquette est considérée comme jetable, c'est-à-dire que la technologie informatique employée pour la réaliser n'est pas forcément assez robuste et évolutive pour être intégrée telle quelle. Pensez à l'analogie de la maquette d'avion qui est très utile en soufflerie, mais qui ne peut pas voler !

Dans un premier temps, les besoins ont été modélisés au moyen des cas d'utilisation UML. Ils seront maintenant représentés de façon plus concrète par une maquette d'IHM (Interface Homme-Machine) destinée à faire réagir les futurs utilisateurs. Les figures (Maquette de la page "Ajouter un élève") (Maquette de la page "rechercher élève") montrent les maquettes que nous avons réalisé.

On a essayé de respecter les règles et les standards d'IHM des pages web à savoir : une entête qui se compose d'un logo cliquable renvoyant au menu principal, un menu horizontal, un menu vertical, on a aussi représenté des informations sur l'utilisateur avec la possibilité de se déconnecté à partir de n'importe quelle page de notre application, un « fil d'Ariane » (Breadcrumb) [une aide à la navigation sous forme de signalisation de la localisation de l'internaute dans une page d'un site web. Leur but est de donner aux utilisateurs un moyen de garder une trace de leur emplacement à l'intérieur d'un site web], et un pied de page.



Maquette de la page "Ajouter un élève"



Maquette de la page "rechercher élève"

