

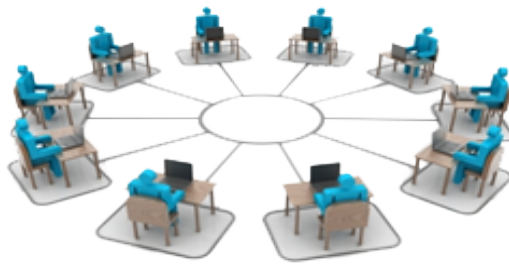
**Mémoire de Projet de Fin d'Année**

*Pour l'Obtention du Diplôme BAC+4 en*

**Ingénierie des Systèmes d'Informatiques**

**Sujet**

**Gestion d'un Centre de Formation**



**Soutenu par :**  
**Mr. Ech-Charafi Yassine**  
**Mr. Hammani Mourad**

**Sous la direction de :**  
**Mr. Zellou Ahmed**

Année Universitaire 2012-2013

# ***Dédicace***

***Nous dédions ce travail à :***

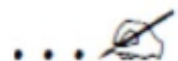
*A Nos chers parents que nous remercions de tous nos cœurs pour  
leurs conseils qui nous ont guidés tout au long de notre chemin  
d'étude;*

*A Notre encadrant qui nous as énormément aidé et soutenu tout au  
long de la réalisation de travail.*

*A Nos enseignants pour leurs formations, leurs conseils et leurs aides.*

*A Nos amis pour leurs encouragements et soutien.*

*A ceux qui nous ont indiqués la bonne voie en nous rappelant que la  
volonté fait toujours les grandes personnes.*



# Remerciements

**« Si j’ai vu plus loin, c’est en me tenant sur les épaules des géants qui m’ont précédé »**  
*Isaac Newton*

Nous tiendrons à remercier dans un premier temps, Monsieur **Ahmed ZELLOU** pour l’aide et les conseils fournis lors des différents suivis se rapportant aux missions évoquées dans ce rapport.

Nous remercions également toute l’équipe pédagogique de SUPMIT et tous les intervenants professionnels pour avoir assuré la partie théorique de notre formation.

Nous remercions dans un dernier temps nos ami(e)s et toutes les personnes qui ont contribué de près ou de loin à la conception et au bon déroulement de ce travail.

# Résumé

Ce document présente notre Projet de Fin d'Année dans le cadre du diplôme BAC+4 en Ingénierie des Systèmes d'Informatiques de SUPMTI. Il a pour objectif d'appliquer nos acquis et connaissances.

Il nous a été confié la réalisation d'une application Web offrant des services de Gestion d'un centre de formation.

Ce projet est destiné aux bénéficiaires désirant faire l'inscription en ligne dans des différentes formations, il propose une solution complète :

- Une application évoluée basée sur un modèle métier pour la présentation des formations.
- La gestion et mise en relation entre les formateurs, bénéficiaires et les formations.
- L'ajout et la gestion des thèmes, catalogues et les formations.

Pour la réalisation de ce projet, nous avons adopté le processus de développement 2TUP qui fait une large place à la technologie et à la gestion du risque, nous avons aussi utilisé le langage UML pour la modélisation.

En ce qui concerne le développement, nous avons utilisé le langage Php5, JavaScript et Ajax et MySQL pour la base de données.

Dans le présent document nous allons décrire plus en détail les étapes que nous avons réalisées dans le cadre de notre projet.

Mots-clés : Centre de formation, 2TUP

# Abstract

This document presents our Final Project Year in the degree BAC +4 Engineering of Information Systems SUPMTI. It aims to apply our learning and knowledge.

We have been entrusted the creation of a website offering services management training center.

This project was designed for beneficiaries who wishing to register online in different formations, it offers them a complete solution:

- An evolved web site based on a business model for the presentation of formations.
- The management and interaction between trainers, beneficiaries and formations.
- The insertion and management themes, catalogs and formations.

For this project we adopted the process 2TUP development has a strong focus on technology and risk management, we also used the UML for modeling.

Concerning the development, we used Php5, JavaScript and Ajax and MySQL for the database.

In this paper we will describe in more detail the steps we have made in our project.

## Liste des abréviations

Abréviation	Désignation
<b>2TUP</b>	2 Track Unified Process
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>BPMN</b>	Business Process Model and Notation
<b>CSS</b>	Cascading Style Sheets
<b>DOM</b>	Document Object Model
<b>FTP</b>	File Transfer Protocol
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>IHM</b>	Interface Homme Machine
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>MSP</b>	Microsoft Project
<b>MVC</b>	Model View Controller
<b>OMG</b>	Object Management Group
<b>PBS</b>	Product Breakdown Structure
<b>PHP</b>	Hypertext Preprocessor
<b>RUP</b>	Rational Unified Process
<b>SQL</b>	Structured Query Language
<b>UML</b>	Unified Modelling Language
<b>XML</b>	Extensible Markup Language
<b>XP</b>	Extreme Programming
<b>XSL</b>	Extensible Stylesheet Language

# Table des figures

Figure 1: Le processus de développement en Y.....	17
Figure 2: L'organigramme PBS .....	19
Figure 3: Planning prévisionnel du projet.....	20
Figure 4: Branche fonctionnelle du processus Y .....	28
Figure 5: Diagramme de cas d'utilisation.....	30
Figure 6: Diagramme de séquence (Authentification) .....	31
Figure 7: Diagramme de séquence (Ajouter Formation) .....	32
Figure 8: Diagramme de séquence (S'inscrire) .....	33
Figure 9: Diagramme de séquence (S'inscrire dans formation) .....	34
Figure 10: Diagramme de séquence (S'inscrire comme Formateur).....	35
Figure 11: Branche technique du processus Y .....	37
Figure 12: Architecture 3 tiers .....	39
Figure 13: Architecture du désigne pattern MVC.....	44
Figure 14: Phase de mise en œuvre dans le processus Y .....	47
Figure 15: Architecture logicielle du projet.....	48
Figure 16: Diagramme de classes.....	52
Figure 17: L'interface d'accueil de l'application.....	53
Figure 18: L'interface d'authentification .....	54
Figure 19: L'interface de création d'un profil formateur .....	54
Figure 20: L'interface d'ajout d'une formation .....	55
Figure 21: L'interface de lister et gérer des bénéficiaires.....	55

# Liste des tableaux

Tableau 1: Tableau comparatif entre les processus de développement .....	16
Tableau 2: Liste des acteurs .....	26
Tableau 3: Les vues de l'application .....	49
Tableau 4: Description de la classe formateur .....	51
Tableau 5: Description de la classe formation .....	51



# Table des matières

Introduction générale.....	11
Présentation de la structure d'accueil.....	14
1.1 Mission.....	14
1.2 Description.....	14
Contexte du projet.....	15
Conduite du projet.....	15
1.3 Processus de développement.....	15
1.4 Langage de modélisation.....	18
1.5 Planification du projet.....	18
1.5.1 Structure de répartition du produit (PBS).....	19
1.5.2 Diagramme de Gantt.....	19
Conclusion.....	20
1 Cahier des charges.....	22
1.1 Introduction.....	22
1.2 Présentation générale du projet.....	22
1.3 Recueil des besoins fonctionnels.....	23
1.4 Recueil des besoins opérationnels.....	25
2 Identification des acteurs.....	26
3 Conclusion.....	26
1 L'étude fonctionnelle dans le processus Y.....	28
2 Capture des besoins fonctionnels.....	28
3 Diagrammes de cas d'utilisation.....	29
Diagramme de séquence.....	31
Conclusion.....	35
1 L'étude technique dans le processus Y.....	37
2 Capture de besoin technique.....	37
2.1 Spécification technique.....	38
2.1.1 Spécification d'architecture.....	38
2.1.2 Architecture 3-tiers.....	39
Conception générique.....	40
2.2 Technologies et outils utilisées.....	40
2.2.1 PHP.....	40

2.2.2	JavaScript .....	41
2.2.3	AJAX.....	41
2.2.4	Microsoft Project.....	42
2.2.5	Dreamweaver .....	42
2.2.6	Enterprise Architect.....	43
2.2.7	XAMPP .....	43
2.3	Design patterns .....	44
2.3.1	MVC.....	44
	Conclusion.....	45
1	La mise en œuvre dans le processus Y .....	47
2	La Conception préliminaire.....	47
2.1	Architecture logicielle implémentée .....	48
2.2	Énumérations des vues de l'application .....	48
3	La Conception détaillée.....	49
3.1	Classes métiers détaillées .....	50
3.1.1	Conception des classes .....	50
3.1.2	Conception des attributs .....	50
3.1.3	Description des classes .....	50
3.1.4	Diagramme de classes .....	52
4	Réalisation .....	53
4.1	Implémentation des IHM .....	53
4.1.1	L'interface d'accueil de l'application .....	53
4.1.2	L'interface d'authentification.....	54
4.1.3	L'interface de création d'un profil formateur .....	54
4.1.4	L'interface d'ajout d'une formation.....	55
4.1.5	L'interface de lister et gérer des bénéficiaires .....	55
	Conclusion générale et perspective .....	57
	Bibliographie et Webographie .....	58

## Introduction générale

Aujourd'hui, vu l'intérêt croissant de vouloir gagner en temps, de conserver les données en toute sécurité, de limiter le nombre d'employés et bien d'autres raisons, ont poussé petites, moyennes et grandes entreprises à chercher des solutions informatiques capables de répondre à leurs besoins.

Dans ce cadre s'inscrit notre projet de fin d'année qui consiste de réaliser une solution web destinée à la gestion des formations des différents thèmes.

La solution à mettre en place sera composée d'un modèle métier pour la présentation du catalogue des formations, la gestion des bénéficiaires, des formateurs, et les relations entre les différents intervenants.

Afin d'adopter des éléments de réponse à cette demande, le présent mémoire détaille à travers ses différents chapitres la démarche adoptée et les résultats atteints. Le présent rapport comporte quatre chapitres :

Le premier chapitre sera consacré au contexte général du projet, nous présentons l'école SupMTI ainsi nous définissons la conduite à suivre, en choisissant une méthode pour le processus de développement (la méthode en Y), tout en respectant une planification basé sur deux mois.

Dans le deuxième chapitre on va faire une étude préliminaire, on posant le cahier des charges tout en spécifiant les acteurs (Administrateur, Formateur, Bénéficiaire). Enfin, nous déterminons les jalons de la capture des besoins du système.

Dans le troisième chapitre qui est consacré à l'étude fonctionnelle, nous nous intéressons à la capture des besoins fonctionnels, ensuite nous réalisons la description des cas d'utilisation et la description de séquence des scénarios les plus importants (ajouter formateur par exemple).

Au niveau du quatrième chapitre, nous allons aborder d'une part la description de la spécification technique par le choix d'un style d'architecture en 3-tiers, et d'autre part, les technologie et les outils utilisées comme PHP, JavaScript, Ajax, Dreamweaver, Xampp...en utilisant la méthode d'architecture MVC.

Le cinquième chapitre concrétisera le modèle de conception entamé lors des précédents chapitres, surtout en des couches applicatives et des interfaces homme/machine (interface d'ajout d'une formation par exemple).

Dans la conclusion, nous synthétisons le travail réalisé et énumérons les perspectives envisagées (développement d'une solution de E-Learning par exemple).

# Chapitre 1

---

## Contexte général du projet

Ce chapitre présente le sujet du projet. Et précise l'objectif et la conduite du projet ainsi que le planning général de la réalisation.

## Présentation de la structure d'accueil

Ecole supérieure de Management, d'Informatique et de Télécommunication

### 1.1 Mission

L'école supérieure de Management, d'Informatique et de Télécommunication, SupMIT, a pour mission d'assurer une formation d'excellence aussi bien théorique et pratique dans les domaines liés au management et des systèmes d'information et de télécommunication.

L'école SupMIT est convaincue qu'une formation de qualité passe par une compréhension et une proximité du monde économique. Cette proximité favorise l'innovation scientifique et la qualité pour une insertion réussie dans le monde du travail. de part l'expérience de son équipe pédagogique, SupMIT représente un pont entre les savoirs de l'université et les attentes opérationnelles des entreprises.

Tout au long du cursus, sont offerts aux futurs lauréats :

- Des enseignements d'ouverture sur le monde économique, depuis les jeux de simulation et de gestion d'entreprise jusqu'aux cours de sensibilisation à l'éthique industrielle.
- Des stages obligatoires dans le monde professionnel.

### 1.2 Description

Pour répondre aux exigences d'un enseignement de haut niveau la quasi-totalité des professeurs SupMIT sont titulaires d'un Doctorat ou d'un PhD et issus des grandes écoles : ENSIAS, INPT, EMI, ISCAE, ENSG.)

Dotés d'expériences professionnelles en entreprise ou en sociétés de conseil, remarquées pour leurs travaux de recherche et leurs publications au niveau national et international, ils apportent une réelle valeur ajoutée aux enseignements qu'ils dispensent.

Plusieurs enseignants sont certifiés sur plusieurs nouvelles technologies. Ils préparent ces étudiants à la certification sur des produits en forte demande sur le marché :

CISCO

Microsoft

HP

## Contexte du projet

Notre projet a été réalisé au sein de l'école SupMIT, il a pour but de développer une application d'inscription en ligne des formations variées au domaine informatique.

Cette application contient des différents modules :

- **Module Administrateur** : c'est lui qui fait la totalité de la gestion d'application, parmi ces missions gestion des formations, gestion des bénéficiaires, gestion des formateurs.
- **Module Bénéficiaire** : c'est un acteur majeur pour l'application, il peut s'inscrire sans réserver une formation ou bien il peut s'inscrire au moment de la réservation. Après la réservation, il a le droit de modifier ou supprimer une réservation.
- **Module Formateur** : sa principale mission est d'assisté une formation après l'inscription, ou bien l'inscription sans engager une formation.

Tous ces acteurs ont besoin d'authentifier avant chaque opération.

## Conduite du projet

Afin de mieux organiser le déroulement des activités du projet, nous avons suivi une démarche de gestion de projet pour aboutir aux objectifs en organisant le déroulement temporel des tâches, à partir de l'élaboration du cahier des charges à la réalisation du produit souhaité.

### 1.3 Processus de développement

En général, la gestion d'un projet informatique consiste à choisir un processus de développement adéquat. Ce dernier constitue un facteur déterminant dans la réussite d'un projet, du fait qu'il cadre ses différentes phases et caractérise les principaux traits de sa conduite.

Pour cela le choix d'une méthode de développement adéquate aux exigences fonctionnelles, techniques et qualitatives d'un projet doit être élaboré au préalable afin d'obtenir un produit

qui répond aux besoins et attentes des utilisateurs.

Devant une multitude de méthodologies disponibles, le choix devient difficile, Pour cela, nous étions amenées à effectuer un «benchmarking» entre les principaux processus.

Le **tableau 1** suivant présente une étude comparative entre différents processus :

	Description	Points Forts	Points Faibles
<b>RUP</b>	RUP est à la fois une méthodologie de conduite et de développement de projets et un outil prêt à l'emploi	Propose des modèles de documents, et des canevas pour des projets types	Coûteux à personnaliser. Très axé processus, au détriment du développement : peu de place pour le code et la technologie
<b>XP</b>	XP est une méthode agile plus particulièrement orientée sur l'aspect réalisation d'une application, sans pour autant négliger l'aspect gestion de projet	Préparation des phases de vérification au moment de l'Analyse et de la Conception	Obligation de la présence du client Développement en binôme.
<b>2TUP</b>	2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels	Fait une large place à la technologie et à la gestion du risque	Ne propose pas de documents types

**Tableau 1: Tableau comparatif entre les processus de développement**

Nous constatons que toutes ces méthodologies proposent de travailler de façon itérative, que ce soit au niveau des plannings, des spécifications ou du développement. Si l'itératif s'est imposé, c'est parce qu'il réduit la complexité de réalisation des phases en travaillant par approches successives et incrémentales.

Il est alors possible de présenter rapidement aux utilisateurs des éléments de validation. De plus, l'itératif permet une gestion efficace des risques en abordant, dès les premières itérations, les points difficiles. Le choix de la technologie et de l'architecture logicielle et applicative de notre futur système est l'une des phases les plus importantes de notre projet vu le degré élevé de sa complexité technique. Ainsi 2TUP semble être le mieux adapté pour mener à bien notre projet, puisqu'il réserve une branche entière aux aspects techniques dans le processus de développement.

Le processus 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Illustré en **figure 1**, le processus en Y s'articule autour de 3 branches [Roques & Vallée, 2007] :



- Une branche fonctionnelle
- Une branche technique
- Une phase de réalisation

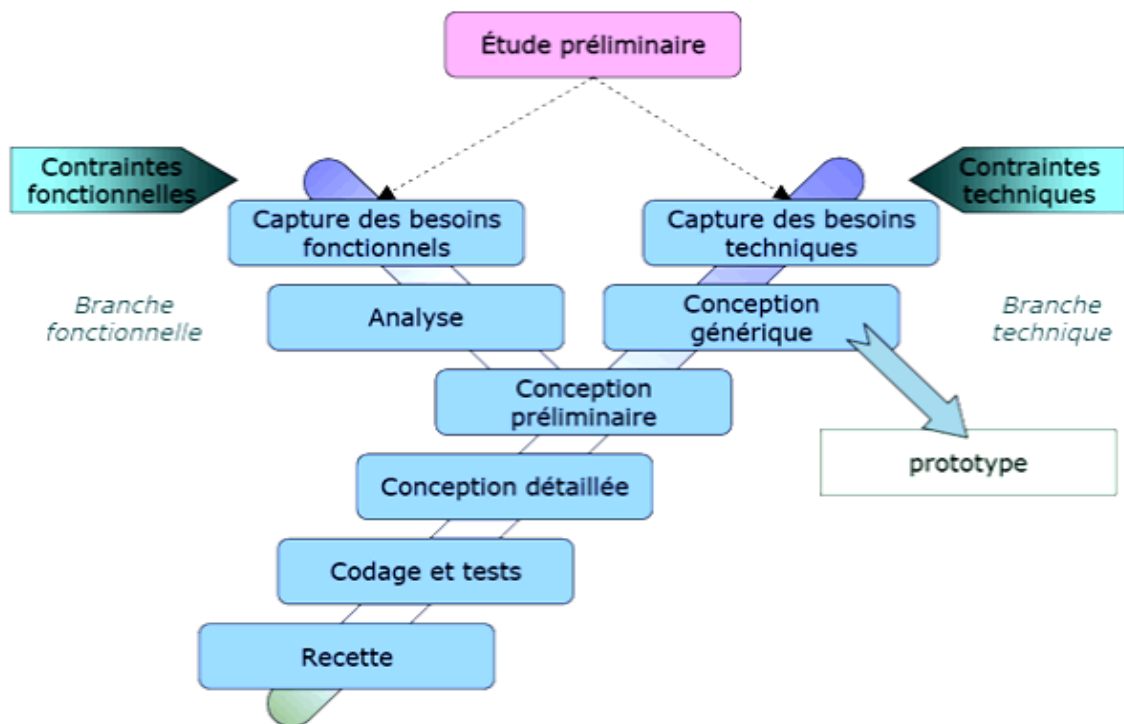


Figure 1: Le processus de développement en Y

La branche fonctionnelle capture des besoins fonctionnels pour élaborer un modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité de l'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

Quant à la branche technique, elle permet de rassembler les besoins techniques (sécurité, montée en charge, intégration à l'existant...). Elle propose également une architecture logicielle et applicative qui répond aux contraintes présentées dans le dossier technique. Et enfin elle permet de proposer des règles de développement afin d'industrialiser l'implémentation (gestion des exceptions, règles de nommage, règles de codage, ...).

La troisième branche comporte la conception qui intègre le modèle d'analyse et de l'architecture technique puis étudie comment réaliser chaque composant. Elle comporte aussi l'étape de codage qui produit ces composants et teste les unités de codes réalisées et l'étape de validation qui consiste enfin à valider les fonctions du système développé.

## 1.4 Langage de modélisation

UML (langage de modélisation unifié) est un concept permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

UML se décompose en plusieurs sous-ensembles :

- Les vues : Les vues sont les éléments observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues il est possible de définir (ou retrouver) le système complet.
- Les diagrammes : Les diagrammes sont des graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
- Les modèles d'élément : Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types.

## 1.5 Planification du projet

Conduire un projet, c'est assurer le pilotage d'un processus de changement avec des ressources dédiées en optimisant les compétences, l'organisation, les systèmes et les outils de conduite.

Une approche managériale réactive, souple et systématique pour mener à bien des changements importants, complexes, ciblés sur le but à atteindre, il y a trois niveaux de gestion de projet : la gestion de la production, des ressources et du temps. A fin de satisfaire à ce dernier critère qu'est la gestion du temps, il est nécessaire d'établir un planning prévisionnel. Sachant que la ressource affectée à ce projet est l'élève ingénieur et que le présent planning est réajusté au fur et à mesure de l'avancement du projet.

La première étape de notre projet a été d'analyser des besoins fonctionnels du système en étudiant les spécifications de notre projet. Cette étape nous a permis de dégager les besoins fonctionnels du projet. Ensuite, nous avons effectué des recherches sur l'architecture

logicielle afin de choisir l'environnement de développement le plus adéquat aux exigences techniques de notre application.

La seconde étape a été de proposer la modélisation du système, puis la dernière étape a été de mettre en œuvre l'environnement et implémenter notre application.

### 1.5.1 Structure de répartition du produit (PBS)

Conformément aux exigences fonctionnelles spécifiées dans le cahier des charges, le projet peut être décomposé en cinq grandes phases à savoir : le démarrage et l'initialisation, L'analyse et la conception, le développement, les tests, le déploiement et clôture du projet.

Chaque phase est décomposée en un ensemble d'activités, et donne en résultat un ou plusieurs livrables comme montre la **figure 2** :

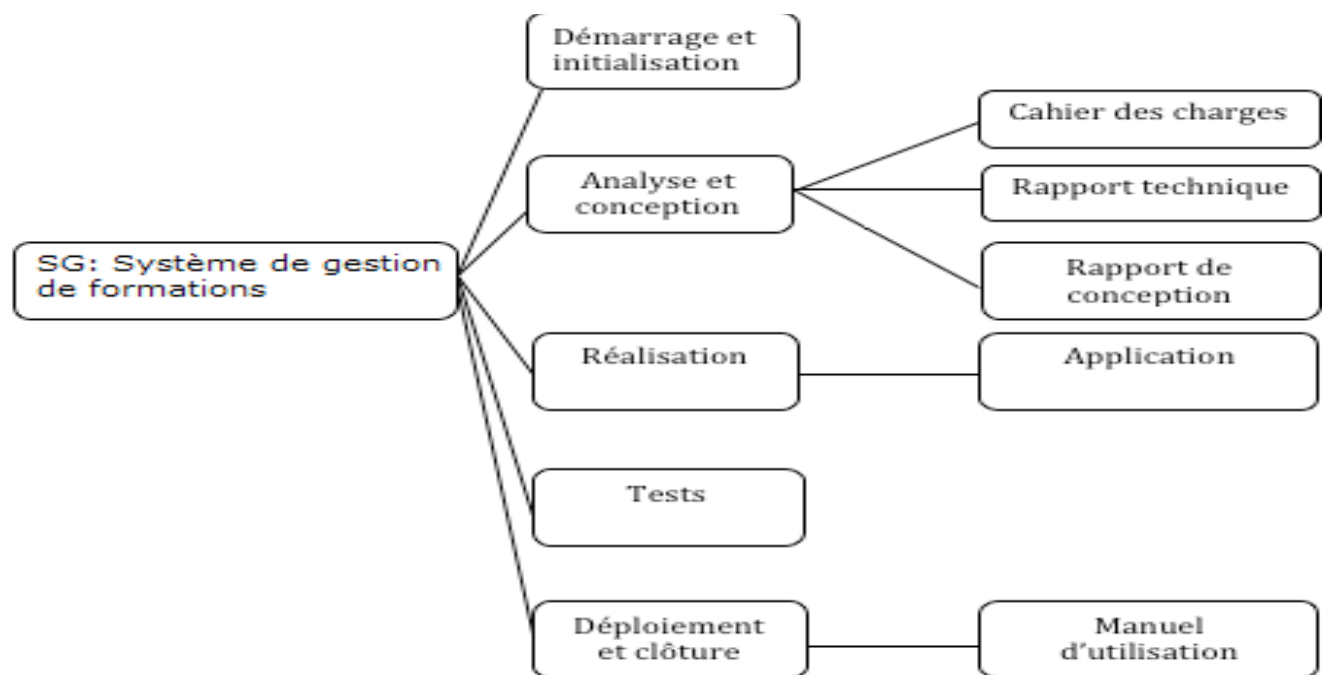


Figure 2: L'organigramme PBS

### 1.5.2 Diagramme de Gantt

La Figure suivante présente sommairement le planning des étapes du déroulement de notre projet **Figure 3** :

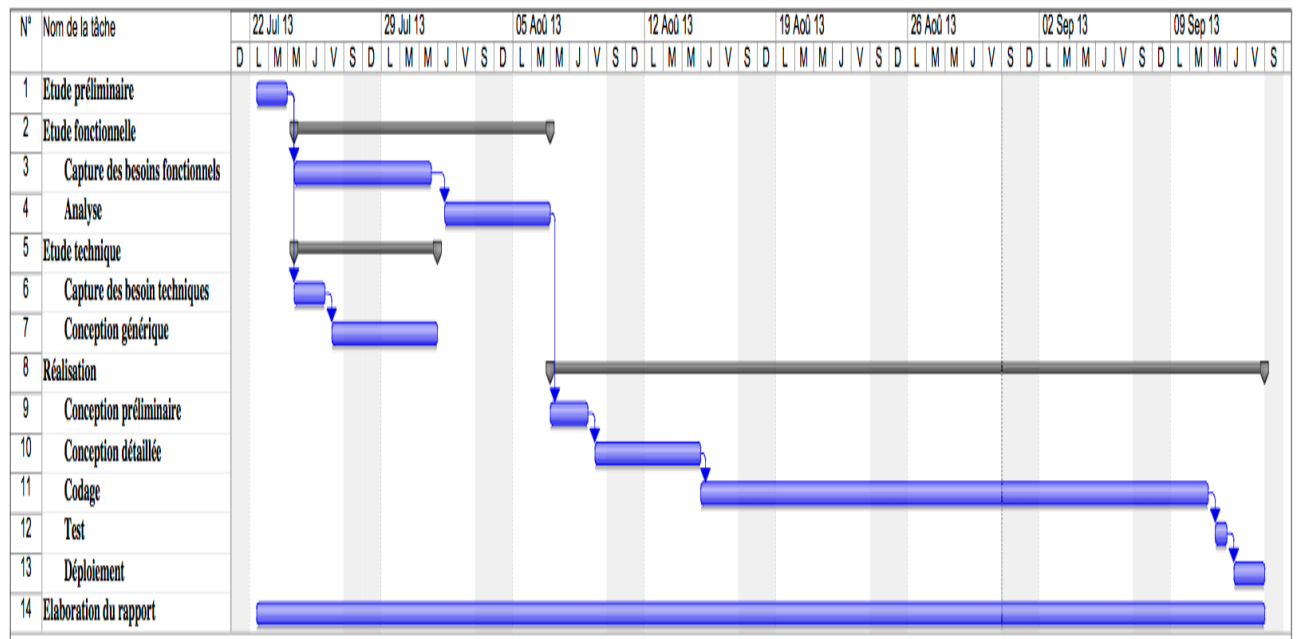


Figure 3: Planning prévisionnel du projet

## Conclusion

Après avoir survolé le contexte général dans lequel s'inscrit notre projet dans cette première partie de ce mémoire, la démarche préconisée pour les réaliser, la description des besoins du projet, l'étude fonctionnelle sera détaillée dans le chapitre suivant.

# Chapitre 2

---

## Etude Préliminaire

Ce chapitre a pour objectif de déterminer les jalons de la capture des besoins du système à mettre en place. Nous commencerons par donner une version textuelle du cahier des charges et ensuite enchaînerons avec un premier modèle contextuel afin d'établir avec précision les limites fonctionnelles du système. Ceci naturellement après identification des acteurs qui interagiront avec le système.

# 1 Cahier des charges

## 1.1 Introduction

L'Ecole SupMIT qui dispose parmi les missions qu'elle s'est astreinte la formation, désire améliorer la qualité des services qu'elle offre. Cette école dans le cadre de l'étude qu'elle a mené, a considéré qu'il y avait un besoin de création et de mise au point une application web d'inscription en ligne aux formations proposées. Notamment, pour offrir un service de qualité et augmenter le rendement de son Département de formation et le fortifier. Par ailleurs, une application d'inscription en ligne permettra à SupMIT de compléter les prestations qu'elle offre aux entreprises et aux clients particuliers.

## 1.2 Présentation générale du projet

L'application devra tout d'abord être extrêmement fiable. En effet, son domaine d'application concerne le cœur de l'activité des établissements des formations professionnelles, et son utilisation quotidienne ne devra pas laisser place à l'éventuel point faible.

L'objectif principal est la gestion des bénéficiaires, des formateurs, des thèmes, des catalogues et des formations.

L'application devra notamment :

- Permettre d'inscrire des nouveaux bénéficiaires et formateurs.
- Permettre de faire la gestion des bénéficiaires et des formateurs.
- Permettre l'ajout et la gestion des thèmes, des catalogues et des formations.
- Permettre d'établir des statistiques relatives aux informations enregistrées.
- Permettre d'impression de certaines pages (attestation, fiche bénéficiaire.....).

Nous devons alors atteindre les résultats suivants :

- Disposer de la base de données représentant la partie stockage des informations concernant les bénéficiaires, les formateurs, et le catalogue des formations.
- Assurer la sécurité de l'application par l'authentification au premier écran, l'enregistrement des informations sur les événements qui ont lieu.
- Disposer d'un module d'administration de l'application permettant la gestion, la mise à jour et la maintenance de l'application.

Ceci est fait en suivant les étapes suivantes :

- Création de planning.
- Elaboration d'un scénario maquette et des esquisses des écrans d'application.
- Conception de la base de données en élaborant les cas d'utilisations, Diagramme de séquence, diagramme de classe.
- Implémentation de la base de données.
- Développement de notre application (codage ou programmation).
- Validation de l'application par des scénarios de tests d'exécution.

### 1.3 Recueil des besoins fonctionnels

Un premier tour d'horizon des besoins exprimés par la direction de *SupMIT* a permis d'établir le cahier de charges préliminaire suivant :

- **La création du compte formateur**

Cette phase est obligatoire pour tous les formateurs qui veulent donner des formations. Lorsqu'un formateur veut participer a une formation, il doit commencer par créer un compte suite à cette opération un profil sera créé en remplissant le formulaire de création de compte qui contient les champs suivantes : CIN, Civilité, Nom, Prénom, Mot de passe, Date de Naissance, Lieu de Naissance, Email, Niveau d'Etude, Expérience, Téléphone, Adresse, CV, Photo.

Après la création, ce compte sera vérifié selon les critères suivants :

- Validation de l'adresse mail : pour bloquer les robots de messagerie.
- Vérifier que la création d'un compte avec un mot de passe vide est rejetée
- Vérifier qu'on ne peut créer qu'un seul compte unique.

Après l'attachement à une formation, le formateur a le droit de consulter la liste des bénéficiaires qui les forment.

- **La création du compte bénéficiaire**

Cette phase est très importante pour les bénéficiaires qui veulent s'inscrire à l'application pour avoir le droit de participer à une ou plusieurs formations.

Lorsqu'un bénéficiaire veut utiliser l'application comme moyen de recherche de formations désirées, il doit commencer par la création d'un compte. Lors de cette étape, il saisit des informations suivantes :

CIN, Nom, Prénom, Mot de passe, Sexe, Date de Naissance, Lieu de Naissance, Email, Niveau d'Etude, Téléphone, Adresse, CV, Photo.

Les bénéficiaires peuvent faire des recherches pour découvrir les formations disponibles.

#### ▪ **Gérer les comptes formateurs et bénéficiaires**

L'administrateur a le droit de gérer les comptes formateurs et bénéficiaires comme suit :

- Permettre de modifier le profile formateur et bénéficiaire.
- De lister les comptes des formateurs et des bénéficiaires inscrits.
- Désactiver ou supprimer un compte.
- Inscrire un bénéficiaire dans une formation.
- Affecter un formateur à une formation créée.

#### ▪ **Gestion des profils**

Cette opération est consacrée aux formateurs et aux bénéficiaires pour avoir la capacité de gérer leurs propres profils (consultation, modification et la suppression d'un profil).

#### ▪ **Gestion des thèmes, catalogues et formations par l'administrateur**

L'administrateur a la possibilité de gérer des thèmes, catalogues et formations, c'est-à-dire l'ajout, la mise à jour et la suppression.



## 1.4 Recueil des besoins opérationnels

- **Sécurité**

Pour se connecter au système, chaque utilisateur doit fournir son login et son mot de passe. En cas de succès de connexion, le système devra restituer le rôle de l'utilisateur afin de déterminer les fonctionnalités auxquelles il a droit.

- **Disponibilité**

Le système doit être en permanence à la disposition de ses utilisateurs.

- **Fiabilité**

Le système doit exécuter les fonctions attendues avec la précision requise.

- **Performance**

Le système devra être capable de répondre aux requêtes des différents utilisateurs en un temps raisonnable.

- **Intégrité des données**

Le système à mettre en place devra garantir l'intégrité des données en toute circonstance.

Le recueil des besoins étant réalisé, nous pouvons passer maintenant à la description du Contexte du système. Cette description s'étale sur trois activités qui sont :

- L'identification des acteurs,
- Le recensement des messages,
- La production des diagrammes de contexte




## 2 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages éventuellement porteurs des données.

Après la phase d'analyse, nous avons identifié trois catégories d'utilisateurs :

Les bénéficiaires, les formateurs et les administrateurs.

Le tableau suivant présente les trois acteurs identifiés et le rôle de chacun d'eux **Tableau 2**:

Acteur	Type	Description du Rôle
 <b>Administrateur</b>	Humain	Prépare les sessions et met en place les formations selon un calendrier. Gère les bénéficiaires, et les formateurs.
 <b>Formateur</b>	Humain	Utilise le système pour déposer son CV et voir la liste des bénéficiaires.
 <b>Bénéficiaire</b>	Humain	Consulte les formations, réserve les cours selon les périodes disponibles via une interface conviviale. Il peut aussi voir ses notes.

**Tableau 2: Liste des acteurs**

## 3 Conclusion

Dans ce chapitre, nous avons fait le recueil initial des besoins fonctionnels et opérationnels. Nous avons ensuite identifié les acteurs et ses rôles dans le système.

Le chapitre suivant sera consacré à l'étude fonctionnelle de notre projet.

# Chapitre 3

---

## Etude Fonctionnelle

Nous présentons dans ce chapitre la méthodologie utilisée pour l'analyse et la conception de la base de données et les diagrammes qu'on a pu établir.

# 1 L'étude fonctionnelle dans le processus Y

La branche fonctionnelle **Figure 4** du processus Y a pour objectif la capture des spécifications et des contraintes fonctionnelles. Durant l'analyse, nous nous concentrons sur le métier indépendamment de toute technologie, et nous essayons d'adapter le plus possible notre système aux besoins des utilisateurs.

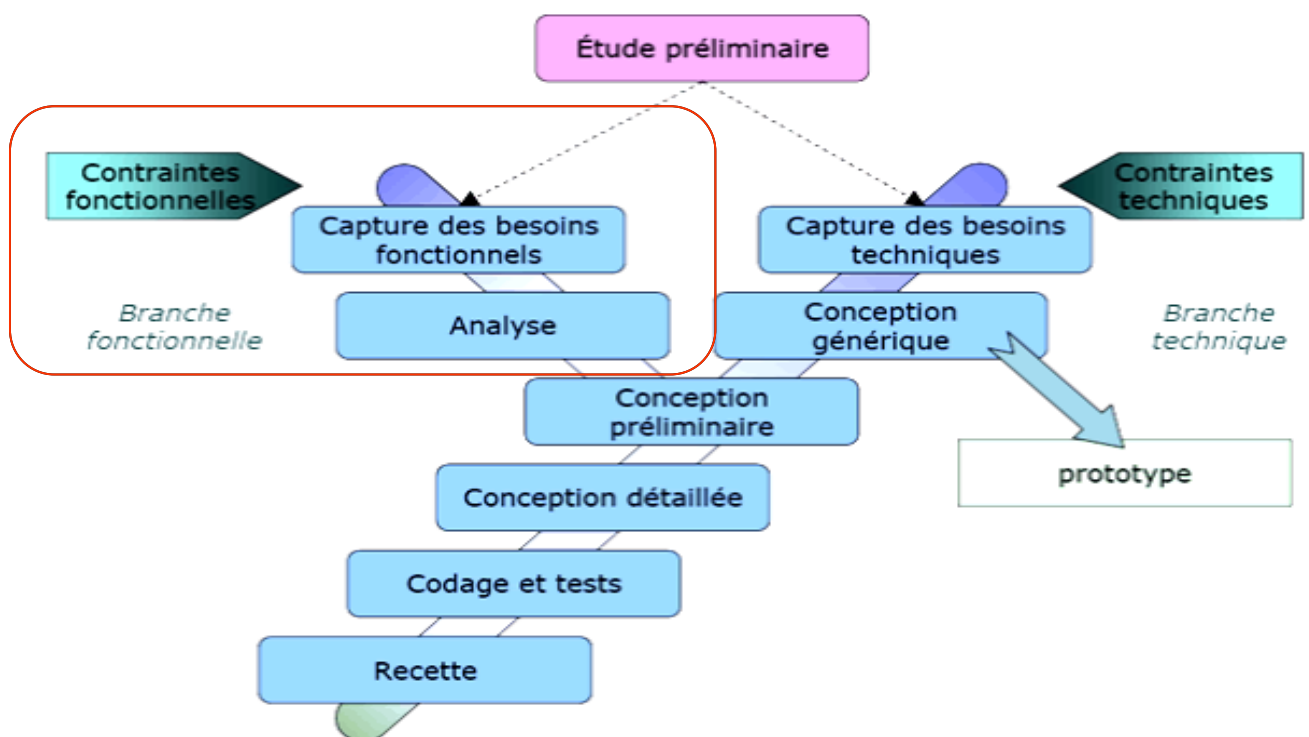


Figure 4: Branche fonctionnelle du processus Y

## 2 Capture des besoins fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Dans cette section, nous allons formaliser et détailler ce qui a été ébauché au cours de l'étude préliminaire.

**La capture des besoins fonctionnels a pour objectif :**

- L'identification des cas d'utilisation du système par ses acteurs.
- La description des cas d'utilisation.
- L'identification des classes des modèles d'analyse.

### **3 Diagrammes de cas d'utilisation**

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner Une vision globale du comportement fonctionnel d'un système logiciel. Un cas D'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou Machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

UML définit une notation graphique pour représenter les cas d'utilisation. Cette notation est appelée diagramme de cas d'utilisation. UML ne définit pas de standard pour la forme Écrite de ces cas d'utilisation. En conséquence, il est aisé de croire que cette notation Graphique suffit à elle seule pour décrire la nature d'un cas d'utilisation. Dans les faits, une notation graphique ne peut donner qu'une vue générale simplifiée d'un cas ou d'un ensemble de cas d'utilisation. Les diagrammes de cas d'utilisation sont souvent confondus avec les cas d'utilisation. Bien que ces deux concepts soient reliés, les cas d'utilisation sont bien plus détaillés que les diagrammes de cas d'utilisation.

La **figure 5** décrit les différents cas d'utilisations de chaque acteur. Elle permet de distinguer les rôles et les acteurs du projet.

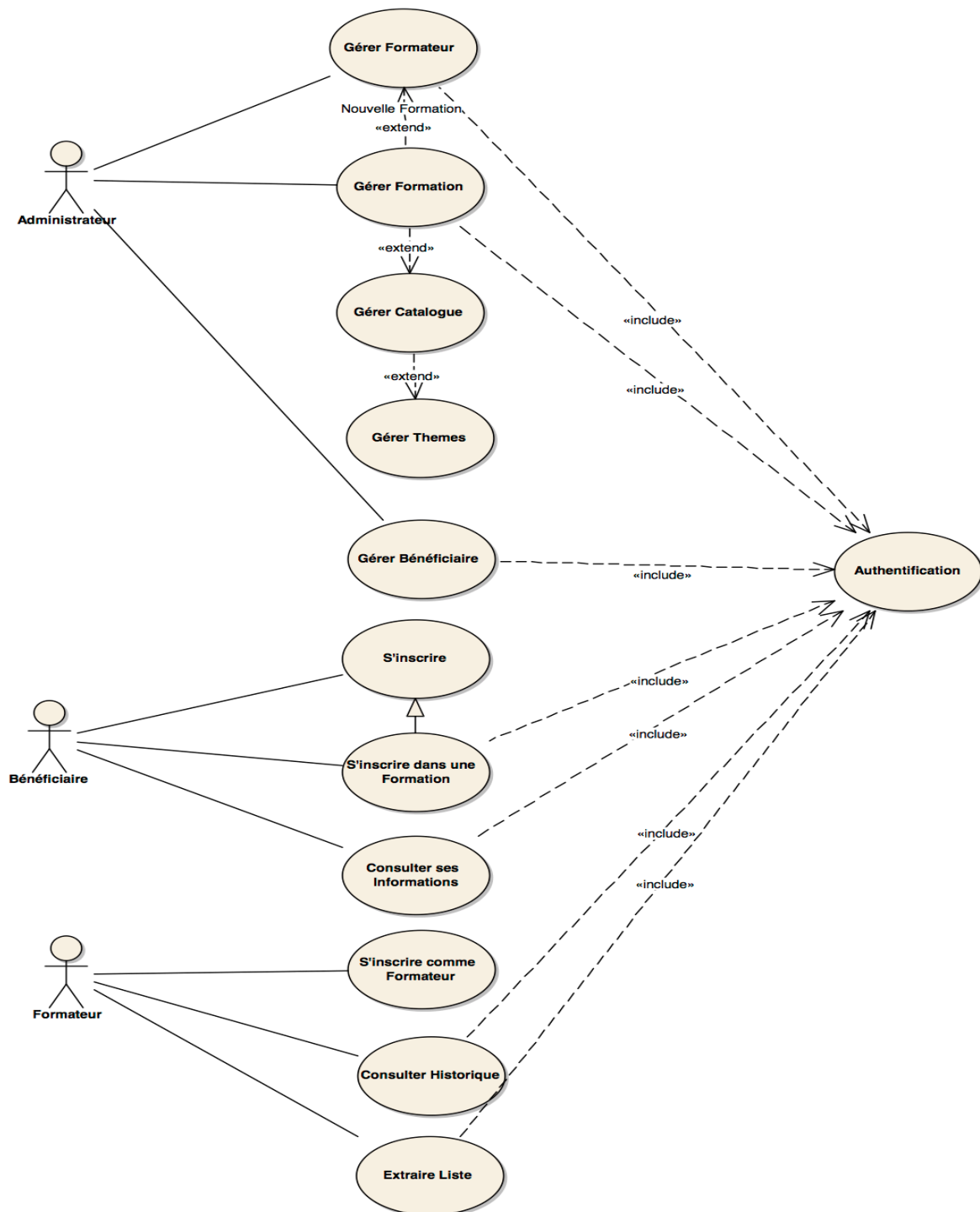


Figure 5: Diagramme de cas d'utilisation

## Diagramme de séquence

Les diagrammes de séquence sont couramment utilisés par un bon nombre d'acteurs d'un projet. En effet, le diagramme de séquence est une représentation intuitive lorsque l'on souhaite concrétiser des interactions entre deux entités (deux sous-systèmes ou deux classes d'un futur logiciel).

Ils permettent à l'architecte/designer de créer au fur et à mesure sa solution. Cette représentation intuitive est également un excellent vecteur de communication dans une équipe d'ingénierie pour discuter cette solution.

Les diagrammes de séquence peuvent également servir à la problématique de test. Les traces d'exécution d'un test peuvent en effet être représentées sous cette forme et servir de comparaison avec les diagrammes de séquence réalisés lors des phases d'ingénierie.

Les diagrammes de séquence tels que définis en UML souffraient cependant d'un gros inconvénient. La quantité de diagrammes à réaliser pouvait atteindre un nombre élevé dès lors que l'on souhaitait décrire avec un peu de détail les différentes branches comportementales d'une fonctionnalité.

La **figure 6** montre la séquence d'authentification. C'est la première séquence déclenchée dans ce projet. L'administrateur entrain de se connecter attend la réponse pour être redirigé vers la page d'accueil correspondante à son profil.

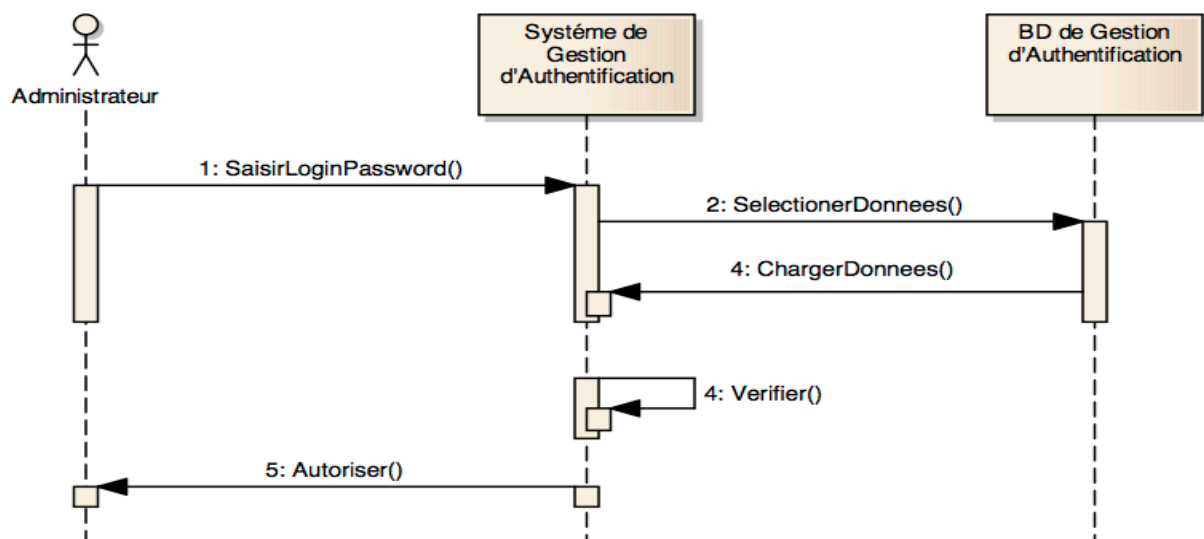


Figure 6: Diagramme de séquence (Authentification)

La **figure 7** décrit la façon avec laquelle un administrateur crée une formation. Il doit spécifier

le catalogue et vérifier la disponibilité de formateur spécialiste.

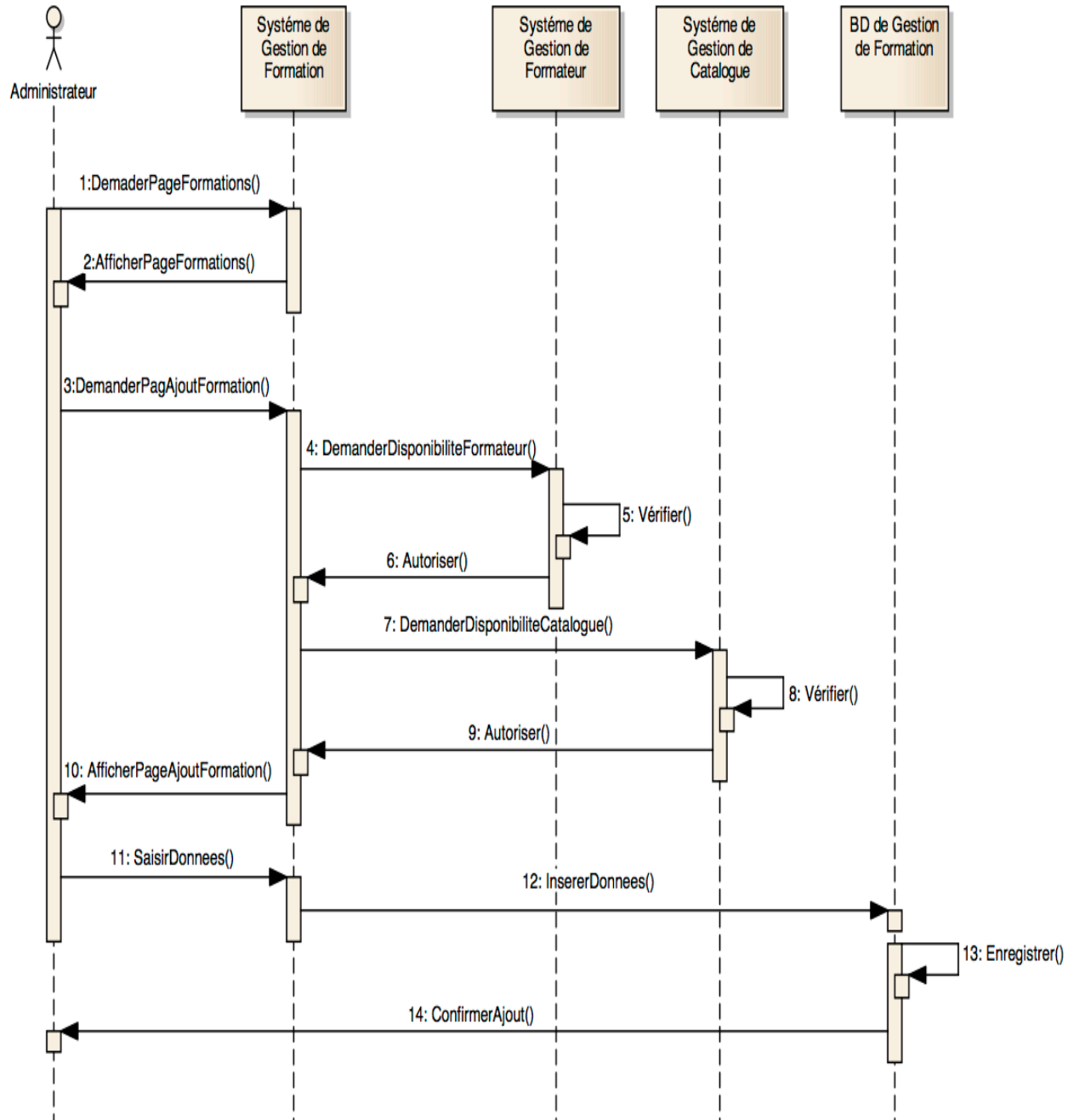


Figure 7: Diagramme de séquence (Ajouter Formation)

La **Figure 8** démontre le processus suivi par le bénéficiaire pour s'inscrire dans l'application.



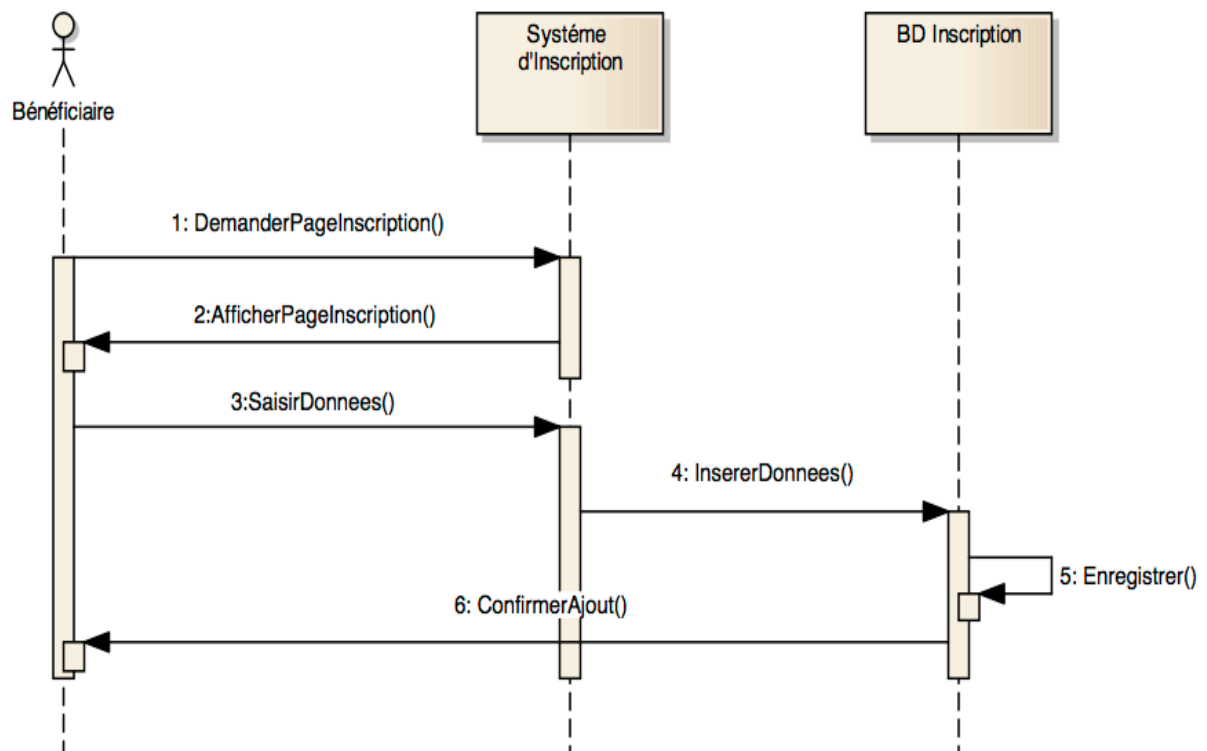
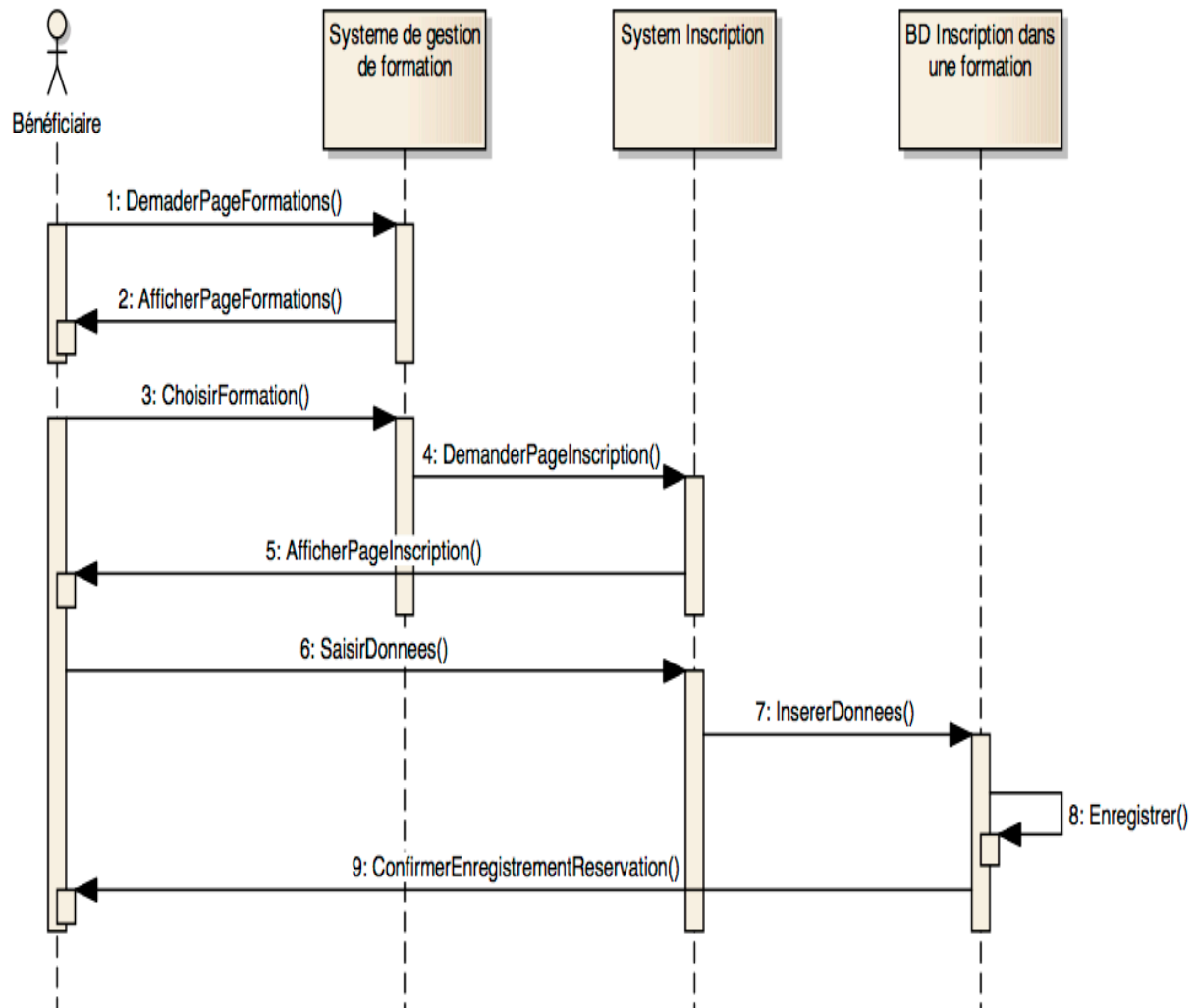


Figure 8: Diagramme de séquence (S'inscrire)

La **Figure 9** démontre le processus suivi par le bénéficiaire pour s'inscrire dans une formation.



**Figure 9: Diagramme de séquence (S'inscrire dans formation)**

La **Figure 10** démontre le processus suivi par le Formateur pour s'inscrire dans l'application en choisissant les formations à encadrer.

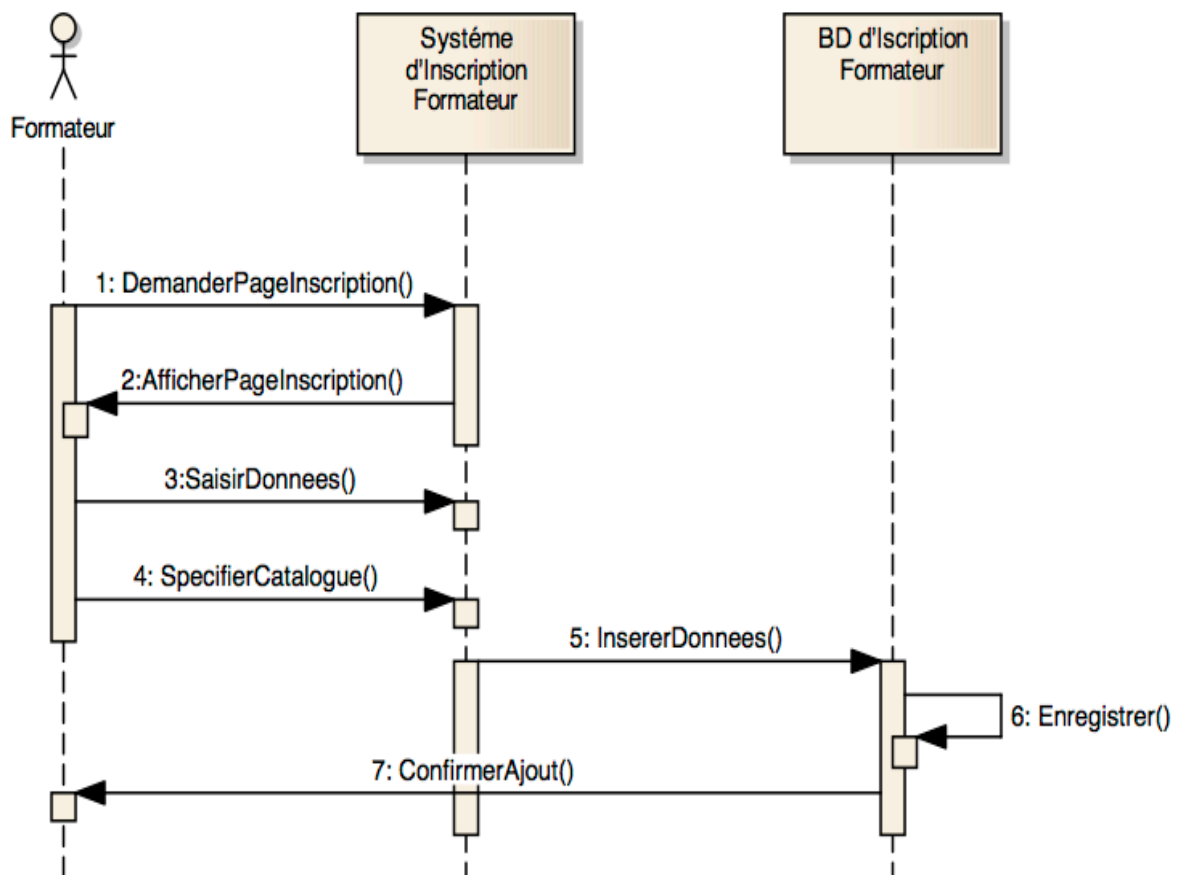


Figure 10: Diagramme de séquence (S'inscrire comme Formateur)

## Conclusion

Dans ce chapitre, nous avons abordé la description des cas d'utilisation et la description des diagrammes de séquence des scénarios les plus importants. Le chapitre suivant sera consacré à l'étude technique de du projet.

# Chapitre 4

---

## Etude Technique

Ce chapitre est consacré conformément au cycle de développement en Y, à la branche technique. Dans cette partie, nous allons aborder la capture des besoins techniques, L'architecture 3-tiers, les designs patterns, les systèmes et les outils de travail utilisés.

# 1 L'étude technique dans le processus Y

L'objectif de la branche technique du processus Y **Figure 11** est de recenser les contraintes et les choix techniques et matériels dimensionnant la conception du système. Le but est d'éviter les risques techniques et les problèmes que l'application pourrait rencontrer tels que les problèmes d'intégration, de maintenance, d'évolution...

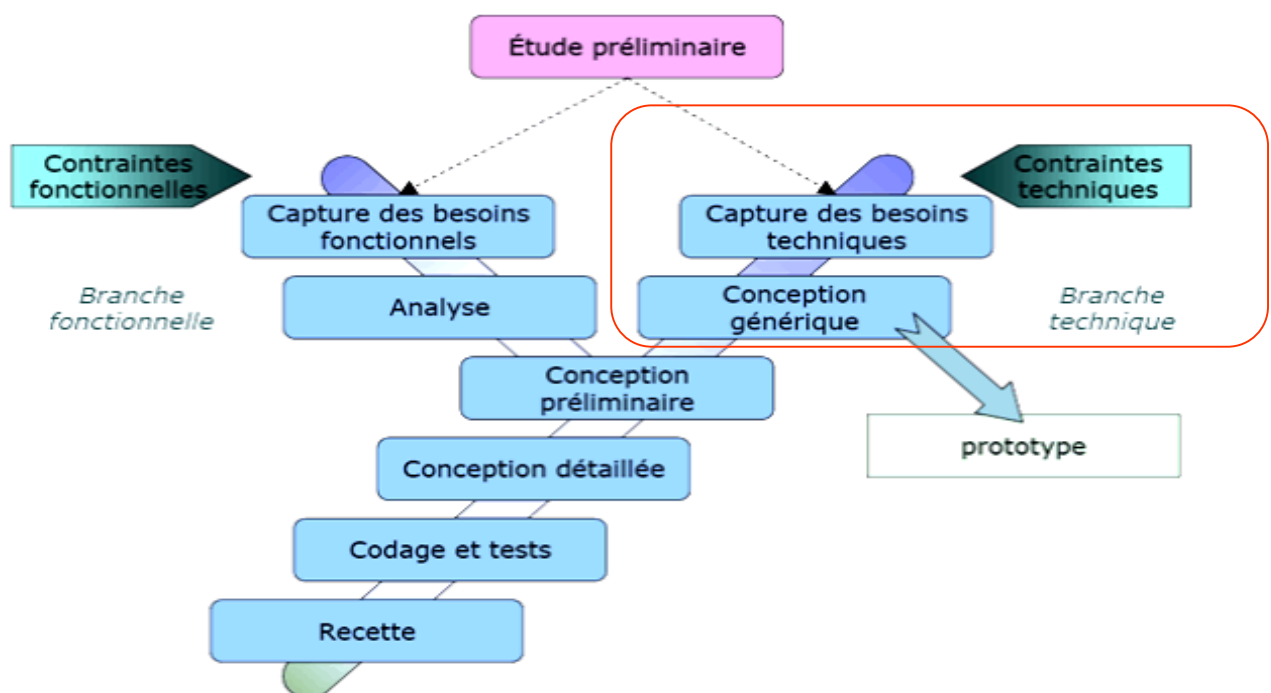


Figure 11: Branche technique du processus Y

## 2 Capture de besoin technique

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. La spécification technique est une activité primordiale pour la conception d'architecture.

## 2.1 Spécification technique

Les prérequis techniques ont été exprimés dans l'étude préliminaire, lors de l'expression des besoins opérationnels. Elles concernent la performance d'accès aux données, la sécurité du système, l'interopérabilité, l'intégrité des données, la disponibilité, et la fiabilité.

### 2.1.1 Spécification d'architecture

L'expression des pré requis techniques implique également le choix d'un style d'architecture client/serveur. Ce choix conditionne la façon dont seront organisés et déployés les composants d'exploitation du système.

- **Composant d'exploitation et composant métier**

- Un composant d'exploitation est une partie du système logiciel qui doit être connue, installée, déclarée et manipulée par les exploitants du système. Un composant d'exploitation doit être interchangeable entre différentes versions et peut être arrêté ou démarré séparément. Il assume des fonctions bien identifiées dans le système, de sorte qu'en cas de dysfonctionnement, le composant incriminé est facilement repérable.
- Un composant métier est un composant d'exploitation dont la fonction est de distribuer les services d'un ou de plusieurs objets métier de l'entreprise.

L'intégration de composants métier implique le recours à un style d'architecture 3-tiers.

- **Style d'architecture en tiers**

Le style d'architecture en tiers (tier signifie «partie» en anglais) spécifie l'organisation des composants d'exploitation mis en œuvre pour réaliser le système. Chaque partie indique une responsabilité technique à laquelle souscrivent les différents composants d'exploitation d'un système. [Roques & Vallée, 2007]

On distingue donc plusieurs types de composants en fonction de la responsabilité technique qu'ils jouent dans le système. Un système client/serveur fait référence à au moins deux types de composants, qui sont les systèmes de base de données en serveur, et les applications qui en exploitent les données en client.

Le style d'architecture 2-tiers correspond à la configuration la plus simple d'un système client/serveur. Dans ce cas, il incombe aux clients de gérer l'interface utilisateur et les processus d'exploitation. Les serveurs ont pour responsabilité de traiter le stockage des données.

L'intégration des objets métier sous la forme de composants métiers fait passer l'architecture client/serveur du 2-tiers au 3-tiers, car elle implique un nouveau type de composants d'exploitation qui s'insère entre les clients et les serveurs de données.

L'architecture 3-tiers **Figure 12** a été davantage choisie parce qu'elle permet de satisfaire les exigences techniques que nous avons repris au niveau de la spécification des besoins techniques à savoir essentiellement :

- ✓ **La sécurité** : elle permet de la définir indépendamment de service et de chaque niveau.
- ✓ **La performance** : elle est assurée du fait du partage des tâches entre les différents serveurs.
- ✓ **La disponibilité** : elle est assurée pour les mêmes raisons que la performance



**Figure 12: Architecture 3 tiers**

### 2.1.2 Architecture 3-tiers

L'architecture adoptée pour notre application est celle la plus utilisée dans le développement des grandes applications, c'est l'architecture 3-tiers. Elle permet de rendre les trois couches (présentation, métier et accès aux données) indépendantes les unes des autres grâce aux interfaces.

- **La couche Présentation**

Elle représente le premier niveau de l'architecture et est la partie visible de l'application et interactive avec l'utilisateur. Dans notre cas, elle est représentée en HTML et est exploitée par des navigateurs Web.

- **La couche Métier**

C'est la couche du second niveau de l'architecture. Elle représente la partie fonctionnelle de l'application. Cette couche opère sur les données en fonction des requêtes des utilisateurs effectuées au travers de la couche présentation (Couche intermédiaire).

- **La couche accès aux données**

La fonction principale de cette couche est de gérer les données. La façon dont elle organise, manipule et stocke les données est transparente vis-à-vis des applications externes et des utilisateurs.

## Conception générique

Elle consiste à développer la solution qui répond aux spécifications techniques que nous avons présentées à l'étape capture des besoins techniques. Cette conception est qualifiée de générique car elle est entièrement indépendante des aspects fonctionnels spécifiés au niveau de la branche de gauche. La conception générique reste donc une activité de la branche de droite.

La standardisation des techniques de développement, à laquelle nous avons assisté ces dernières années, rend cette étape moins conséquente qu'avant. En effet, la diffusion de composants gratuits, tels que Dreamweaver, XAMPP, propose en quelque sorte des architectures logicielles clés en main, et généralement de qualité, qu'il suffit de réutiliser. C'est pour cette raison que nous allons focaliser sur l'exigence technique principale de notre projet, nous avons cité les technologies, Framework et composants utilisés.

## 2.2 Technologies et outils utilisés

Pour satisfaire les besoins fonctionnels et techniques énumérés dans l'étude préliminaire, nous avons optés pour les technologies suivantes :

### 2.2.1 PHP

PHP est un langage de programmation. Sa principale application se situe au niveau de la gestion des sites web dynamiques.

Les capacités de PHP ne s'arrêtent pas à la création de pages web. Il est aussi possible de manipuler des images, de créer des fichiers PDF, de se connecter à des bases de données ou



des serveurs LDAP, et même d'instancier des objets Java. Un module annexe lui permet également de fournir des interfaces graphiques classiques (client lourd, sans navigateur ou serveur web), via GTK.

PHP est à l'origine un langage de script conçu spécifiquement pour agir sur les serveurs web. En ajoutant quelques lignes de PHP à une page HTML, le serveur exécute les instructions correspondantes pour écrire du code HTML à la place. Le résultat (le code HTML initial ajouté à celui produit par PHP) est envoyé au navigateur. Cela permet par exemple d'afficher la date du jour à un endroit bien précis du visuel. On parle alors de page dynamique.

PHP dispose de près de 3 000 fonctions utilisables dans des applications très variées et couvre pratiquement tous les domaines en rapport avec les applications web. PHP 5 et ses nouveautés propulsent PHP dans le monde des plates-formes d'entreprises comme .Net ou J2EE.

### 2.2.2 JavaScript

C'est un langage de script incorporé dans un document HTML. Historiquement il s'agit du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML pour exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. JavaScript a été mis au point par Netscape en 1995. A l'origine, il se nommait LiveScript et était destiné à fournir un langage de script simple au navigateur Netscape Navigator 2. Il a, à l'époque, longtemps été critiqué pour son manque de sécurité, son développement peu poussé et l'absence de messages d'erreur explicites rendant difficile son utilisation. Le 4 décembre 1995, suite à une association avec le constructeur Sun, Netscape rebaptise son langage JavaScript (un clin d'œil au langage Java développé par Sun). A la même époque, Microsoft mit au point le langage Jscript, un langage de script très similaire.

Ainsi, pour éviter des dérives de part et d'autre, un standard a été défini pour normaliser les langages de script, il s'agit de l'ECMA 262, créé par l'organisation du même nom (ECMA, European Computer Manufacturers Association).

### 2.2.3 AJAX

Ajax désigne une approche innovante dans la conception de pages web dont l'objectif est d'optimiser leur interactivité et leur confort d'utilisation conjointe dans les pages web de différentes technologies. Ainsi AJAX incorpore :

- le XHTML et les feuilles de style CSS
- le JavaScript
- le DOM
- l'Object XMLHttpRequest
- le XML et le XSL

### 2.2.4 Microsoft Project

Microsoft Project (ou MS Project ou MSP) est un logiciel de gestion de projets édité par Microsoft. Il permet aux chefs de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets.

Microsoft Project permet la planification des projets, c'est-à-dire la création d'un plan. Il permet la création de tâches et de jalons, leur hiérarchisation, et de définir des liens entre les tâches. Une estimation de la durée et de la charge (ou travail) nécessaire à la réalisation de chaque tâche peut ensuite être réalisée.

Microsoft Project permet la gestion des ressources de chaque projet, c'est-à-dire la création de l'équipe projet puis l'affectation des ressources définies.

Microsoft Project offre une palette de possibilités d'analyse des données du projet et propose de nombreux rapports. Il est même possible d'exporter les informations du projet dans Microsoft Excel ou Microsoft Visio pour analyser le travail et les coûts du projet en fonction de différents axes d'analyse (tâches, ressources, affectation, temps), via des tableaux, graphiques et diagrammes croisés dynamiques.

Microsoft Project permet de communiquer les informations des projets : copie du diagramme de Gantt, impression et surtout, depuis la version 2010, possibilité de créer une frise chronologique exportable vers Microsoft PowerPoint ou dans un message électronique.

### 2.2.5 Dreamweaver

Dreamweaver est un éditeur de site web WYSIWYG pour Microsoft Windows, et Mac OS X créé en 1997, commercialisé par Macromedia puis Adobe Systems sous licence utilisateur final.

Dreamweaver fut l'un des premiers éditeurs HTML de type « tel affichage, tel résultat », mais également l'un des premiers à intégrer un gestionnaire de site (CyberStudio GoLive étant le premier). Ces innovations l'imposèrent rapidement comme l'un des principaux éditeurs de site web, aussi bien utilisable par le néophyte que par le professionnel.

Dreamweaver offre deux modes de conception par son menu affichage. L'utilisateur peut choisir entre un mode création permettant d'effectuer la mise en page directement à l'aide d'outils simples, comparables à un logiciel de traitement de texte (insertion de tableau, d'image, etc.). Il est également possible d'afficher et de modifier directement le code (HTML ou autre) qui compose la page. On peut passer très facilement d'un mode d'affichage à l'autre, ou opter

pour un affichage mixte. Cette dernière option est particulièrement intéressante pour les débutants qui, à terme, souhaitent se familiariser avec le langage HTML.

Dreamweaver a évolué avec les technologies de l'internet. Il offre aujourd'hui la possibilité de concevoir des feuilles de style. Les liaisons avec des bases de données ont également été améliorées ainsi que le chargement des fichiers sur les serveurs d'hébergement. Il propose en outre l'utilisation de modèles imbriqués de pages web, selon un format propriétaire.

### 2.2.6 Enterprise Architect

Outil de Modélisation UML / BPMN / SysML et d'ingénierie dirigée par les modèles (Model-Driven) Enterprise Architect est l'un des outils de modélisation UML les plus utilisés dans le monde. Ce succès est dû à plusieurs facteurs : un support complet d'UML 2.4.1, une excellente ergonomie, un niveau de performance élevé, des versions fréquentes et enfin un coût d'acquisition et de maintenance ultra compétitifs.

Enterprise Architect bénéficie d'une communauté très active et de l'approbation des organismes de standardisation tels que l'OMG. Il a fait ses preuves par des résultats exceptionnels sur de nombreux projets.

Fort de 12 ans de développement par Sparx Systems, Enterprise Architect est employé par de nombreuses entreprises dans tous les domaines d'activité ainsi que par différents organismes de normalisation.

### 2.2.7 XAMPP

Xampp est une application libre regroupant en son sein un serveur web apache, un serveur de base de données (MySQL), un serveur FTP mais également PHP qui communique avec le serveur web apache et traite les pages PHP. Vous disposez aussi d'une interface agréable (PHPMyAdmin) pour gérer aisément les bases de données.

#### 2.2.7.1 MySQL

MySQL est un système de gestion de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur. C'est un logiciel développé sous double licence en fonction de l'utilisation qui en est faite: dans un produit libre ou dans un produit propriétaire.

#### 2.2.7.2 Apache

Apache est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le

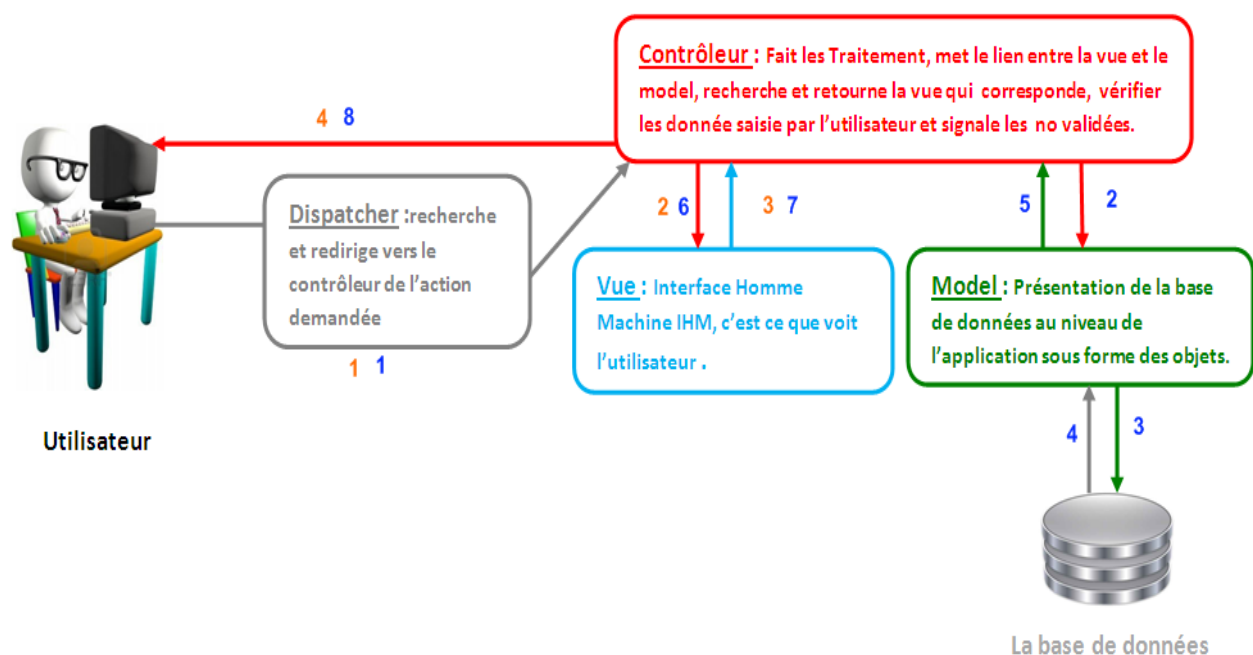
serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc. Néanmoins, il est à noter que l'existence de nombreux modules Apache complexifie la configuration du serveur web.

## 2.3 Design patterns

### 2.3.1 MVC

Pour réaliser notre projet nous avons utilisé les Pattern MVC **Figure 13** :



**Figure 13: Architecture du design pattern MVC**

L'architecture (Modèle Vue Contrôleur) est une méthode de conception pour le développement d'applications logicielles qui sépare le modèle de données, l'interface utilisateur et la logique de contrôle. Ce modèle d'architecture impose la séparation entre les données, les traitements et la présentation, ce qui donne trois parties fondamentales dans l'application finale le modèle, la vue et le contrôleur :

Le Modèle : représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application et définit les méthodes d'accès.

La Vue : correspond à l'interface avec laquelle l'utilisateur interagit. Les résultats renvoyés par le modèle sont dénués de toute présentation mais sont présentés par les vues. Plusieurs vues peuvent afficher les informations d'un même modèle. Elle peut être conçue en html, ou tout autre « langage » de présentation. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle, et de permettre à l'utilisateur d'interagir avec elles.

le Contrôleur : prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle. Il n'effectue aucun traitement, ne modifie aucune donnée, il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande

En résumé, lorsqu'un client envoie une requête à l'application, celle-ci est analysée par le contrôleur, qui demande au modèle approprié d'effectuer les traitements, puis renvoie la vue adaptée au navigateur, si le modèle ne l'a pas déjà fait. Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées. Comme exemple, « Swing » la bibliothèque graphique de Java pour la création d'interfaces graphiques, est basée sur le modèle MVC.

## Conclusion

Dans ce chapitre, nous avons abordé la description des spécifications techniques, l'architecture logicielle et applicative.

Le chapitre suivant sera consacré à la phase de Mise en Œuvre de la solution.

# Chapitre 5

---

## Mise en Œuvre de la solution

L'objectif de ce chapitre qui constitue la dernière branche du processus de développement 2TUP est de concrétiser le modèle de conception présenté lors des précédents chapitres en des couches applicatives et des interfaces homme/machine et le tester pour évaluer les objectifs. Dans cette partie, nous allons successivement nous appesantir sur les phases suivantes :

- Conception préliminaire,
- Conception détaillée,
- Codage,
- Tests,
- Déploiement

# 1 La mise en œuvre dans le processus Y

Après les études fonctionnelles et techniques, on aborde la phase de codage et tests pour chaque module. Ensuite, ces modules sont testés unitairement en vue d'être intégrées dans le système tout entier. Une fois cette étape est terminée, le système sera prêt pour l'activité de test. Cette activité a pour but de tester le système fabriqué en implémentation.

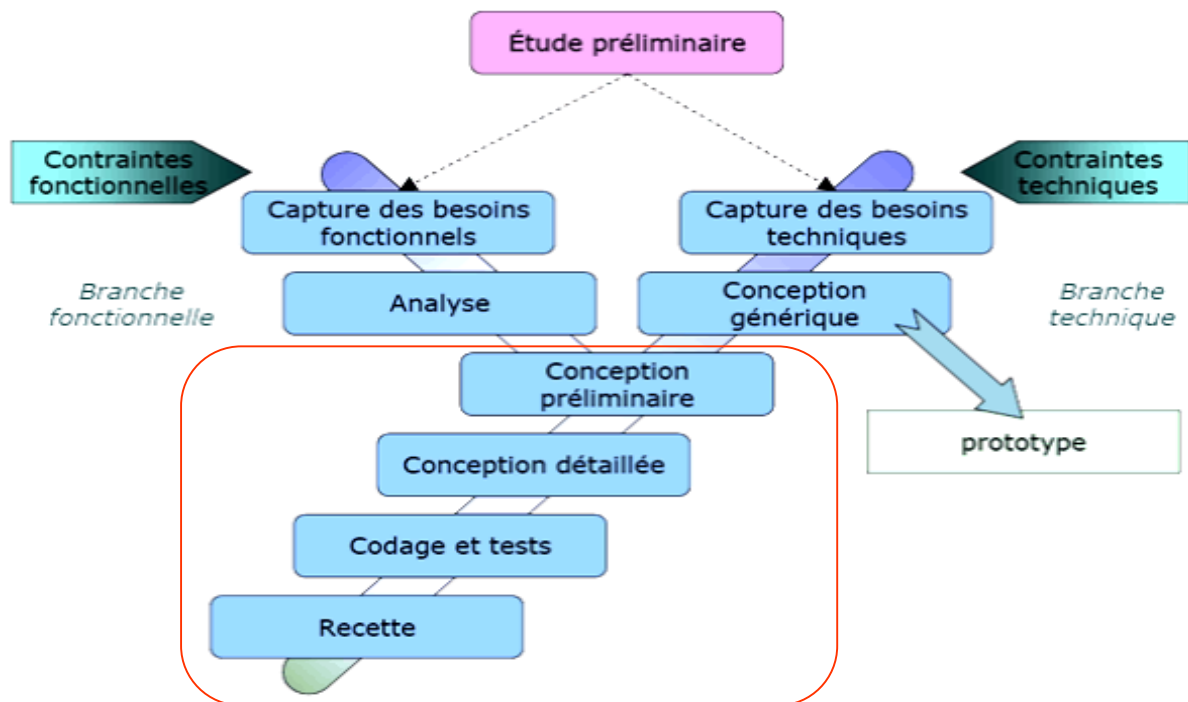


Figure 14: Phase de mise en œuvre dans le processus Y

## 2 La Conception préliminaire

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle en représente le cœur. C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques. En conséquence, plusieurs activités doivent coexister. Il convient de :

- ✓ Passer de l'analyse objet à la conception.
- ✓ Intégrer les fonctions métier et applicatives du système dans l'architecture technique.
- ✓ Adapter la conception générique aux spécifications fournies par l'analyse. La conception préliminaire s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse.

## 2.1 Architecture logicielle implémentée

L'architecture logicielle vient pour démontrer comment le système informatique doit être conçu de manière à répondre aux spécifications de la maîtrise d'ouvrage. L'architecture logicielle implémentée se présente comme suit **Figure 15** :

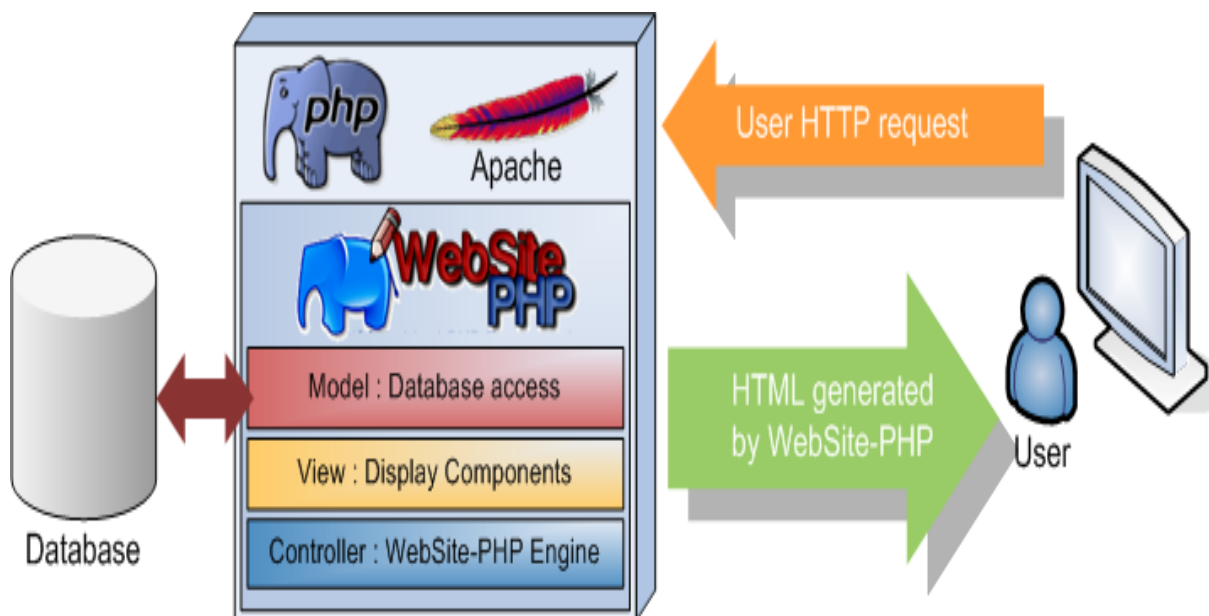


Figure 15: Architecture logicielle du projet

## 2.2 Énumérations des vues de l'application

Cette section consiste à lister les IHM qui définissent les interfaces à travers lesquelles les utilisateurs interagissent avec les différents composants du système.

Nous présentons dans le tableau ci-après, les plus importantes d'entre elles **Tableau 3** :



IHM	Description
L'interface de prise en compte d'un formateur	L'interface d'ajout d'un formateur qui permet à ce dernier de remplir les champs suivants : CIN, Civilité, Nom, Prénom, Mot de passe, Date de Naissance, Lieu de Naissance, Email, Niveau d'Etude, Expérience etc..
L'interface de prise en compte d'un bénéficiaire	L'interface d'ajout d'un bénéficiaire qui permet à ce dernier de remplir les champs suivants : CIN, Nom, Prénom, Mot de passe, Sexe, Date de Naissance, Lieu de Naissance, Email etc..
L'interface d'ajout d'un thème	L'interface d'ajout d'un thème. Dans laquelle l'administrateur saisit le champ suivant : Nom Thème
L'interface d'ajout d'un catalogue	L'interface d'ajout d'un catalogue. Dans laquelle l'administrateur saisit les champs suivant : Désignation, Durée, Description, Coeff_Cours, Coeff_TP parmi les thèmes disponibles
L'interface d'ajout d'une formation	L'interface d'ajout d'une formation. Dans laquelle l'administrateur saisit les champs suivant : Prix, dateDebut, dateFin, Horaire parmi les catalogues disponibles
L'interface de lister et gérer tous les formateurs et bénéficiaires	Cette interface permet de lister et gérer tous les formateurs et bénéficiaires inscrits dans l'application.
L'interface de gérer tous les thèmes, catalogues et formations	Cette interface permet de gérer tous les thèmes, catalogues et formations disponible dans l'application (afficher, modifier ou supprimer)
L'interface d'authentification	Cette interface permet de connecter chaque utilisateur (administrateur, formateur, bénéficiaire) et le redirige vers son propre vue selon son privilège.

**Tableau 3: Les vues de l'application**

### 3 La Conception détaillée

La conception détaillée qui vient juste après est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Le modèle logique y est particulièrement important dans la mesure où c'est dans cette étape qu'on génère le plus grand nombre d'informations. Il est ainsi possible de confier les catégories à des personnes différentes, qui pourront travailler indépendamment les uns des autres. Les concepteurs dans cette phase construisent les classes, les vues d'IHM, les interfaces, les tables et les méthodes qui vont donner une image « prête à coder » de la solution.

### 3.1 Classes métiers détaillées

A partir du modèle logique de conception et des interfaces, le microprocessus de conception de la méthode 2TUP permet de définir les classes à implémenter. Cette activité consiste à :

- **Concevoir les classes** : transformer les classes et méta classes provenant de l'analyse en codage dans un langage de développement.
- **Concevoir les associations** : définir la façon d'exploiter chaque association et les techniques qui seront employées dans le codage.
- **Concevoir les attributs** : identifier les structures de données, les itérations et d'autres types complexes permettant de représenter les attributs d'analyse dans le langage utilisé.

#### 3.1.1 Conception des classes

Les classes issues de l'analyse sont conformes aux possibilités offertes par PHP le langage d'implémentation choisi. Nous n'avons donc pas d'autres classes en dehors de celles provenant de l'analyse.

#### 3.1.2 Conception des attributs

Dans cette section, nous allons définir le type d'attributs identifiés en analyse. Nous pouvons d'ores et déjà constater que tous les attributs des différentes classes se satisfont des types de base de PHP. Concevoir les attributs, c'est également préciser leur visibilité et leur mode d'accès. Pour notre part, les attributs sont privés. Enfin concevoir les attributs c'est aussi spécifier les méthodes qui servent à la mise à jour des attributs.

#### 3.1.3 Description des classes

Dans cette section nous allons décrire quelques classes tableau **4 et 5**:

Classe : Formateur				
Attributs	Visibilité	Nom	Description	Type
	Private	idFormateur	Identifiant de formateur	Entier
	Private	Nom	Nom de formateur	Texte
	Private	Niveau d'Etude	Niveau d'étude de formateur	Texte
	Private	CV	Informations personnels et professionnels de formateur	Texte
Méthodes	Visibilité	Nom	Description	
	Public	displayFormateur()	Permet d'afficher les informations d'un formateur	
	Public	addFormateur()	Permet d'ajouter un formateur	
	Public	updateFormateur()	Permet de modifier les informations d'un formateur	
	Public	deleteFormateur()	Permet de supprimer un formateur existant	

Tableau 4: Description de la classe formateur

Classe : Formation				
Attributs	Visibilité	Nom	Description	Type
	Private	Prix	Prix de la formation	Décimal
	Private	dateDebut	Date de début de la formation	Date
	Private	dateFin	Date de fin de la formation	Date
	Private	Horaire	Horaire d'étude de la formation	Texte
Méthodes	Visibilité	Nom	Description	
	Public	displayFormation()	Permet d'afficher les informations d'une formation	
	Public	addFormation()	Permet d'ajouter une formation	
	Public	updateFormation()	Permet de modifier les informations d'une formation	
	Public	deleteFormation()	Permet de supprimer une formation existante	

Tableau 5: Description de la classe formation

### 3.1.4 Diagramme de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces d'un système ainsi que les différentes relations entre celles-ci. Ce diagramme appartient à la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensembles par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

Le diagramme de classe obtenu après fusion des diagrammes de classes des différentes catégories se présente comme suit **Figure 16** :

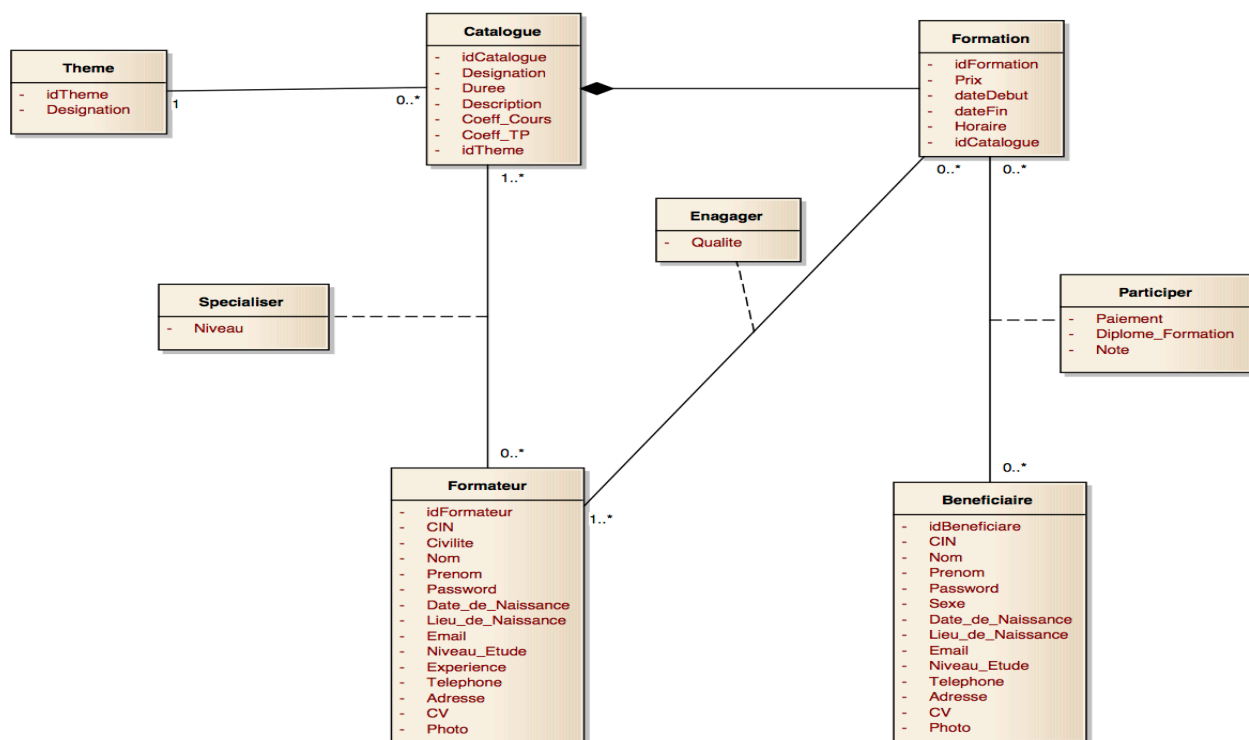


Figure 16: Diagramme de classes

## 4 Réalisation

Dans cette section, nous allons essentiellement nous étaler sur les tâches suivantes :

- ✓ Implémentation des IHM.
- ✓ Codage et tests.
- ✓ Déploiement.

### 4.1 Implémentation des IHM

Nous présentons dans cette section un aperçu de quelques interfaces homme-machines de l'application, résultat de la réalisation des différents modules fonctionnels conçus.

#### 4.1.1 L'interface d'accueil de l'application

C'est la page d'accès par défaut à l'application. Parmi les composants principaux de la page d'accueil:

Zone pour définir l'objectif de l'application.

Zone d'authentification et d'inscription des utilisateurs.

Zone de différentes formations disponibles.



Figure 17: L'interface d'accueil de l'application

### 4.1.2 L'interface d'authentification

A partir de cette interface l'utilisateur peut accéder à son profil.

Figure 18: L'interface d'authentification

### 4.1.3 L'interface de création d'un profil formateur

Cette vue représente le formulaire qui permet aux formateurs de créer un profil formateur sur la plateforme de l'application.

Figure 19: L'interface de création d'un profil formateur

#### 4.1.4 L'interface d'ajout d'une formation

Un formulaire s'affiche lorsque l'administrateur veut ajouter une formation dans la plateforme de l'application.

Figure 20: L'interface d'ajout d'une formation

#### 4.1.5 L'interface de lister et gérer des bénéficiaires

Cette vue permet de lister et gérer tous les bénéficiaires inscrits dans l'application.

Code	Nom	Prénom	Supprimer	Modifier	Afficher
1	Ech-Charafi	Yassine	✗		
2	Ech-Charafi	Soukaina	✗		
3	Hammani	Mourad	✗		
4	Abellaoui	Saadia	✗		
6	Bourich	Hassan	✗		
7	Bourich	Hassan	✗		
9	Ayat	Mohammed	✗		
10	Stitou	Mohammed	✗		
11	Haibour	Azouz	✗		
12	Hmimid	Samir	✗		

Nombre des Bénéficiaires : 10

[Ajout nouveau bénéficiaire](#)

Figure 21: L'interface de lister et gérer des bénéficiaires

## **Conclusion**

Dans ce chapitre nous avons abordé la conception préliminaire à travers la conception détaillée par le biais du diagramme de classes à documenter.

Ce chapitre boucle également le processus 2TUP qui nous avons commencé par l'étude Préalable.



## Conclusion générale et perspective

Jusqu'à présent, l'école SupMIT ne disposait pas de plateforme informatique pour l'inscription en ligne des bénéficiaires et formateurs qui veulent participer à des formations déterminées et utilise principalement des moyens non informatisés pour médiatiser et gérer son travail. C'est dans ce cadre que était s'inscrit notre projet de fin d'année dans lequel, il nous a été demandé de concevoir et réaliser une plateforme sécurisée d'inscription en ligne.

Le système doit permettre aux formateurs de s'inscrire pour être engagé dans des futures formations, et aux bénéficiaires de s'inscrire pour réserver une ou plusieurs formations désirées.

Nous avons pu effectuer, dans les délais, l'analyse et la conception ainsi que la réalisation du Système tout en respectant le cahier des charges initialement déterminé et délimité.

Au cours de ce projet, nous avons eu l'opportunité de mettre en application les différentes connaissances acquises durant nos études. Nous avons également pu approfondir notre savoir en informatique par la découverte de nouvelles technologies.

En perspectives, il y a faire tous ce qui est évaluation de formateurs de la part des bénéficiaires, incluant les commentaires.

Et aussi il reste la deuxième partie de la solution qui est le développement d'une solution de E-Learning dans le quelle les formateurs aurons tous ce qui nécessaire de la technologie pour faciliter les formateurs en ligne et en particulier à travers la présente solution web, en facilitant les procédures de paiement électronique pour les bénéficiaires.

A la fin, nous précisons que la solution conceptuelle et architecturale que nous avons pu mettre en œuvre permettra, sans aucun doute, l'adhésion de nouveaux modules et composants sans pour autant nuire au bon fonctionnement du système.

## Bibliographie et Webographie

### Ouvrage

- |   |  |
|---|--|
| [Roques & Vallée, 2007]<br>4ème édition EYROLLES                  | UML2 en action De l'analyse des besoins à la<br>conception |
| [Eric Daspet & Cyril Pierre de<br>Geyer]<br>4ème édition EYROLLES | PHP 5 avancé□  |
| [Vincent Capitaine, 2010]<br>Dunod, Collection InfoPro            | Project 2010 : guide pratique pour les chefs de projet     |
| [Luc Van Lancker]<br>ENI édition                                  | Ajax : Développer pour le Web 2.0                          |

### Adresse Internet

- |                |   |
|----------------|---|
| [Wikipédia]    | <a href="http://fr.wikipedia.org/wiki/Adobe_Dreamweaver">http://fr.wikipedia.org/wiki/Adobe_Dreamweaver</a>   |
| [Sparxsystems] | <a href="http://www.sparxsystems.fr/">http://www.sparxsystems.fr/</a>   |
| [Site du Zéro] | <a href="http://www.siteduzero.com/informatique/tutoriels/votre-serveur-local/mamp">http://www.siteduzero.com/informatique/tutoriels/votre-serveur-local/mamp</a> |