



MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE Abbès LAGHROUR DE KHENCHELA
FACULTE DES SCIENCES ET DE TECHNOLOGIE



Département de math et informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence (L.M.D)

Spécialité : INFORMATIQUE GENERALE

CRÉATION SITE WEB DYNAMIQUE POUR LA FACULTÉ SCIENCES ET TECHNOLOGIE

Réalisé par :

- *Hoggas Azzeddine*
- *Boughdiri Kheireddine*

Dirigé par :

Dr : CHERGUI.W

Présenté le.../06/2015

Remerciement

En terminant notre mémoire de fin d'études, il nous est agréable d'adresser nos vifs remerciements à notre Allah en premier et dernier temps, à tous ceux qui ont aidé de près ou de loin à élaborer ce mémoire,

Nous remercions en particulier nos encadreurs:
Dr CHERGUI.W

Ainsi que tous professeurs qui ont enseigné durant nos études à la faculté des sciences et techniques/département Math et informatique
À la fin nous tenons à remercier tous nos collègues d'étude, particulièrement notre promotion

Merci à tous

DÉDICACE

Je dédie ce mémoire à :

• *Mes parents :*

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Quisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mes frères et soeurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Azzeddine hoggas

Dédicace

Je dédie ce mémoire à :

Mes parents :

Ma mère BAHIA qui a œuvrée pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour tout son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-elle, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, MED SALAH qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mes frères (KAMEL et DJAMEL) qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Mes amis cursus qui ont m'accompagné durant le mal et le bien, Merci HACHEM ,TAKI ,SALIM et KHALED

Mes professeurs de l'université qui doivent voir dans ce travail la fierté d'un savoir et savoir-faire bien acquis.

Introduction générale

Chapitre 1

1.	
Introduction.....	01
2. Intérêts d'un site web.....	01
3. Classification des sites web.....	01
4. Types des sites web.....	02
5. Notion de webmastering.....	03
6. Cycle de vie d'un site web.....	03

Chapitre 2

1.indroduction.....	08
2.UML (<i>Unified Modeling Language</i>) :	08
3. Le processus unifié :	18

Chapitre 1

1. introduction :	25
2. Exigences fonctionnelles :	25
3. spécification des exigences d'après les cas d'utilisations :.....	26
4. Identification des concepts du domaine :.....	28
5. Diagrammes de séquence système :.....	29
6.Conclusion	35

Chapitre 1

1.	
Introduction :.....	36
2. environnement de développement :.....	36
3. Présentation de l'application :.....	41
4. implémentation et réalisation de l'application web :.....	41
5. Captures.....	43
6. conclusion :.....	46

Conclusion générale

Introduction générale

Au cours de ces dernières années, les nouvelles technologies de l'information et de la communication ont connu un bouleversement marqué par l'apparition de l'Internet et par sa croissance exponentielle.

Ces années ont aussi été marquées par l'entrée en scène de *World Wide Web* (WWW), l'une des applications qui a popularisé Internet, et qui a fait exploser le nombre de ses utilisateurs. Par conséquent, le web suscite l'intérêt de la majorité des organisations qui se doivent enrichir leur présentation via des sites web.

Ainsi les universités prennent plus place dans l'espace numérique de jour en jour. Cette vision d'instauration ne se limite pas à l'utilisation des outils informatiques et la mise à disposition des praticiens, les nouvelles technologies de communication et d'information mais actuellement, les universités profitent du web pour se créer une identité propre.

Ainsi dans un souci continu d'améliorer les méthodologies d'enseignement en favorisant l'accès aux nouvelles technologies de communication et d'information, Notre Faculté a mis en place un site web qui sert comme moyen de communication et qui offre aux étudiants et personnels de la faculté, la possibilité d'accéder à distance à une base de données, de partager avec les enseignants des documents et de profiter de plusieurs autres services. Mais ce site est en cours du développement et il y'a encore quelques fonctionnalités et quelques rubriques inactives.

Le but de notre projet est de participer au développement de ce site en reconstruisant les pages web des trois branches.

Dans ce mémoire, nous présentons la méthodologie de travail avec ce genre de projet. Et nous décrivons les étapes de réalisation ainsi que les résultats qu'on a obtenus.

Dans le premier chapitre, le projet et les besoins sont présentés. Passant de l'état actuel du site, on va définir ce qu'on appelle cahier de charges.

Le deuxième chapitre contient importance Uml dans le développement web .

Dans le troisième chapitre, conception le site

Dans le quatrième chapitre, les techniques des outils qu'on a besoin dans la réalisation de notre projet sont examinées.

CHAPITRE 1

1 Introduction

Un site web est un ensemble de fichiers web (pages web), liés par des liens hypertextes, stockés sur un serveur web, c'est-à-dire un ordinateur connecté en permanence à internet, hébergeant les pages web.

2 Intérêts d'un site web

La mise en place d'un site web peut être motivée par plusieurs raisons :

- Le **besoin de visibilité** : un site web, dans la mesure où il fait l'objet d'une bonne campagne de promotion, peut être un moyen pour une organisation d'augmenter sa visibilité ;
- L'**amélioration de la notoriété** : grâce à un site web institutionnel ou un mini site web événementiel, une enseigne peut développer sa popularité auprès du public ;
- La **collecte de données** : internet représente pour les entreprises une formidable opportunité de recueillir des données sur leurs clients ou bien de démarcher de nouveaux prospects ;
- La **vente en ligne** : les internautes peuvent faire des achats en ligne de certains produits de consommation. Un site internet peut représenter pour certaines entreprises une opportunité en terme de commercialisation ;
- La **mise en place d'un support aux utilisateurs** : de plus en plus de sociétés utilisent internet comme support privilégié pour le service avant-vente ou après-vente. En effet, avec un site web, il est possible de mettre à disposition des internautes un maximum d'informations commerciales, techniques ou institutionnelles, à moindre coût.

3 Classification des sites web

On distingue habituellement plusieurs **catégories** de sites web, selon le but poursuivi :

- Les **sites vitrine** (appelés également *sites plaquette* ou *site identité*) sont des sites dont l'objectif est de mettre en avant l'image de marque de la société, en présentant par exemple ses produits ou ses services ;
- Les **sites catalogue** sont des sites visant à présenter l'offre de l'entreprise ;
- Les **sites d'information** sont des sites fournissant une information particulière à un type d'internautes ;
- Les **sites marchands** sont des sites vendant directement des produits aux internautes et permettant éventuellement de payer en ligne ;

- Les **sites institutionnels** sont des sites présentant l'organisation et ses valeurs. Ce type de site décrit généralement l'activité de l'organisation, des chiffres clés et donne les informations nécessaires aux internautes ;
- Les **sites personnels** (parfois *pages perso*) sont des sites réalisés par des particuliers à titre de loisir, le plus souvent par passion pour un sujet ou une discipline ;
- Les **sites communautaires** sont des sites réunissant des internautes autour d'un intérêt commun.
- Les **sites intranet** sont des sites accessibles de l'intérieur d'une entreprise ou d'une direction, ayant pour objet la mise à disposition et le partage d'informations professionnelles.

4 Types des sites web

Site web statique – Site web dynamique ? C'est en définissant les objectifs de votre projet de création que nous pourrions déterminer précisément le type du site web.

4.1 Création d'un site web statique

Un site web statique est un site composé de quelques pages dont le contenu est écrit directement dans la page. Il faut choisir ce type de site si votre contenu n'a pas besoin d'être mis à jour régulièrement ou si les informations fournies par le site ne risquent pas de changer ou d'évoluer car les sites statiques n'ont pas de Back-office (outil de gestion du site web)



Figure 1.1 : les sites statiques

4.2 Création d'un Site internet dynamique

Un site web dynamique est un ensemble de pages dont le contenu est exclusivement stocké en base de données. Ce type de site web permet la modification du contenu à tout moment et en

toute autonomie. Il peut avoir des galeries de photos, des rubriques d'actualités etc... Grâce à la gestion facile et la flexibilité des modifications, les sites dynamiques représentent un grand pourcentage de sites présents sur internet.



Figure 1.2 : les sites dynamique

5 Notion de webmastering

On appelle **webmaster** une personne en charge d'un site web, c'est-à-dire généralement la personne qui conçoit un site web et le met à jour.

Le terme **webmastering** désigne l'ensemble des tâches nécessaires à l'exploitation d'un site web.

6 . Cycle de vie d'un site web

La vie d'un site web possède deux principales phases, chacune décomposables en étapes spécifiques :

- La **création**, correspondant à la concrétisation d'une idée en un site en ligne, référéncé et visité ;
- L'**exploitation**, correspondant à la gestion quotidienne du site, son évolution et sa mise à jour.

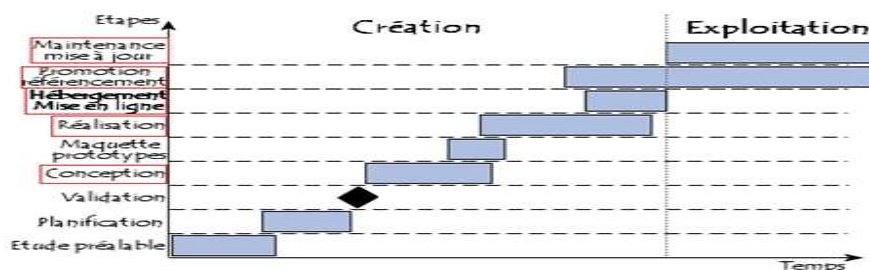


Figure 1.3 : Cycle de vie d'un site web

La phase de **création** d'un site web est un projet à part entière comprenant un grand nombre de phases :

- **Conception**, représentant la formalisation de l'idée ;
- **Réalisation**, correspondant au développement du site web ;
- **Hébergement**, se rapportant à la mise en ligne du site, de manière permanente.

La phase d'**exploitation** du site englobe notamment les activités suivantes :

- **Promotion et référencement**, permettant de développer son audience ;
- **Maintenance et mise à jour**, représentant l'animation quotidienne du site et le maintien de son bon fonctionnement.

La mise en place d'un site web est donc une activité multi-disciplinaire faisant appel à un grand nombre de compétences. Selon l'organisation, la fonction de webmaster pourra relever d'une fonction de chef de projet à celle de l'« homme-orchestre », chargé de toutes les activités, de la conception au référencement.

6.1 Conception

La plupart des ouvrages traitant de la création d'un site internet abordent uniquement la phase de réalisation (création de pages web), essentiellement technique, et délaissent la plupart du temps l'ensemble des étapes d'avant-projet.

La mise en place d'un site web s'agit avant tout d'une démarche stratégique et créative devant être réalisée de manière participative.

6.1.1 Comparaison entre publication papier et publication web

Avant de se lancer dans une publication Web, il est difficile d'éviter la comparaison avec la publication papier.

La plupart des concepts pour structurer l'information s'appuient sur l'organisation de livres, brochures, périodiques ou catalogues qui sont développés autour de la publication imprimée. Culturellement, il est impossible (voire dangereux car vous risquez de décontenancer votre interlocuteur) de passer outre à des siècles de règles et de conventions. Mais force est de constater que la publication sur Internet comporte ses propres spécificités :

- L'internaute n'a pas la sensation physique du livre et risque d'être rapidement perdu dans la présentation de l'écran qui se présente à lui.

- En Html, le langage informatique utilisé pour écrire les publications Web, vous n'avez pas la maîtrise de votre document comme dans une feuille de traitement de texte qui sort de votre imprimante.
- L'internaute a une démarche active vers l'information qu'il recherche.
- La différence essentielle est cependant l'interactivité qu'apporte la publication Web. Prenons simplement les e-mails de vos visiteurs [1 à 2% en moyenne], qui vous communiquent en temps réel, encouragements, critiques ou suggestions.

6.1.2 Schéma global

Avant de se lancer dans la réalisation d'un site Web, il faut prendre le temps d'une réflexion préalable dont voici les lignes de force.

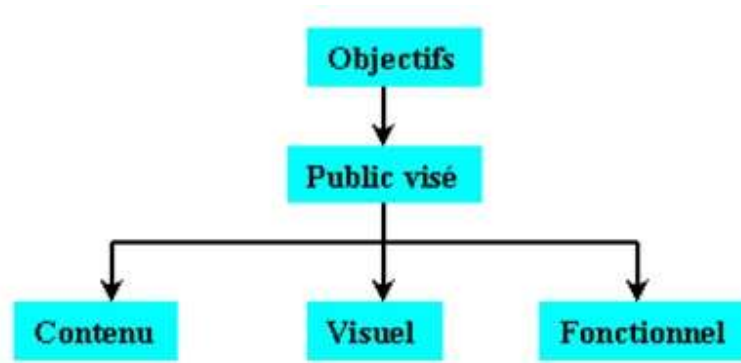


Figure 1.4 : Schéma de site web

"Un site Web est donc trois composantes qui s'adressent chacune à une fraction de votre public.

- Ceux qui n'ont jamais visité votre site, les passants, apprécieront sa clarté et sa présentation visuelle harmonieuse.
- Ceux qui y reviennent une ou deux fois, les curieux, seront convaincus par la qualité du contenu.
- Enfin les habitués des lieux, les fidèles, vous témoigneront une infinie gratitude si, en plus, le téléchargement de vos pages est rapide." (Philippe Monteiro da Rocha).

6.2 Réalisation

La réalisation du site concerne la création des pages web et des éléments graphiques. La création des pages web consiste à créer des fichiers web comme par exemple des fichiers HTML. Il existe deux façons de créer ce type de fichier :

- En **éditant les fichiers HTML « à la main »**, c'est-à-dire en saisissant le code HTML dans un fichier texte à l'aide d'un simple éditeur de texte. Cette solution est la plus fastidieuse (malgré la relative simplicité du langage HTML), mais reste toutefois la meilleure façon d'apprendre à réaliser un site, de comprendre comment celui-ci fonctionne, et d'être ainsi en mesure de créer un code optimisé et propre.
- En **utilisant un éditeur HTML WYSIWYG** (*What You See Is What You Get*, traduisez *Ce que vous voyez est ce que vous obtenez*). Il s'agit d'un logiciel permettant de créer des pages web visuellement en plaçant des objets et des contrôles. Le logiciel se charge de générer le code HTML automatiquement. Il s'agit d'une solution très pratique pour créer des pages web, car la complexité est en grande partie masquée. Cette méthode peut s'avérer très ennuyeuse si l'éditeur ne permet pas de réaliser ce que l'utilisateur souhaite. Une connaissance du langage HTML est néanmoins souhaitable afin de pouvoir maîtriser les options d'édition avancées du logiciel, permettant notamment de modifier manuellement des attributs de style.

D'autre part, un site web convivial doit contenir des images. Un logiciel de dessin sera nécessaire afin d'égayer le site avec des images (au format GIF, JPG ou PNG).

6.3 Hébergement

Une société mettant à disposition un serveur web connecté en permanence à internet est appelée **hébergeur** et offre un service appelé **hébergement**. On distingue deux principales catégories d'hébergeurs :

- **les hébergeurs gratuits**. Ils prêtent gratuitement un espace disque sur un serveur pour créer vos pages web. Ce type de service est généralement totalement gratuit. Ils gagnent de l'argent soit avec un espace publicitaire sur votre site (ce procédé se fait de plus en plus rare), soit uniquement grâce au trafic sur leur propre site.
- **les hébergeurs professionnels**. Ce type d'hébergement garantit un service de qualité (bande passante) et de sécurité (sécurité des données et assurance d'un nombre de pannes réduit). Il s'avère assez indispensable dans le cas d'un site à trafic important (plus de 1000 visiteurs par jour) et permet l'achat d'un nom de domaine (un nom du type [www.commentcamarche.net]).

6.4 Mise en ligne

Pour créer un site web, il n'est pas nécessaire d'être connecté à internet, compte tenu du fait qu'il s'agit uniquement dans un premier temps de créer les fichiers HTML et les images.

Toutefois, lorsque le site web est prêt à être vu par des visiteurs, il est essentiel de le mettre en ligne, c'est-à-dire de transférer (copier) les fichiers de votre ordinateur sur le serveur.

Pour ce faire il est alors essentiel de se connecter à internet, puis d'envoyer les pages sur le serveur à l'aide d'un **client FTP** (un logiciel de transfert de fichiers).

CHAPITRE 2

1. introduction

En informatique UML (de l'anglais *Unified Modeling Language*), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement, et en conception orientée objet. UML est couramment utilisé dans les projets logiciels.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.5 bêta 2 depuis septembre 2013

2 .UML (Unified Modeling Language) :

2.1 Utilité d'UML :

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet/logiciel
- Acteurs
- Processus
- Schéma de base de données
- Composants logiciels
- Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés.

2.2 Historique :

UML est le résultat de la fusion de trois de méthodes d'analyse orientées objet : OOD, OMT et OOSE.

La méthode OOD, *Object Oriented Design*, de G.Booch a été conçue à la demande du Ministère de la Défense des USA. L'objectif était de préparer de façon rigoureuse la structuration des programmes écrits en langage ADA ou C++.

La méthode OMT, *Object Modeling Technique*, a été mise au point à General Electric. Ses auteurs ont puisé leur inspiration d'une part dans les langages à objets pour des applications d'informatique industrielle (automates, contrôle de processus...), d'autre part dans les techniques de modélisation conceptuelle des méthodes d'analyse des années 80. OMT

représente un système comme un assemblage d'éléments auxquels on attache des comportements, c'est-à-dire des opérations pouvant être déclenchées à la réception d'un message envoyé par d'autres composants.

La méthode OOSE, *Object Oriented Software Engineering*, est d'origine universitaire (informatique temps réel) et industrielle (Ericsson). Son originalité consiste à faire reposer l'analyse sur une expression par l'utilisateur de la façon dont il pense utiliser le futur système.

Devant l'attentisme du marché face aux méthodes et aux AGL objets, la société Rational Software a réuni les auteurs principaux de ces trois méthodes pour qu'ils se mettent d'accord sur un langage de modélisation dans l'espoir qu'il devienne une référence. Sa réussite fut d'être retenu comme norme de modélisation par l'OMG, après avoir reçu le soutien de plusieurs grands constructeurs informatiques et éditeurs de logiciels. Ce langage a passé par différents stades et est encore en évolution.

Date	Stade d'UML	Acteurs	Action
1991		G.Booch J.Rumbaugh	Elaboration de méthodes orientées objet
1994		Rational Software	Rapprochement de G.Booch et J.Rumbaugh
1995	Méthode unifiée 0.8	G.Booch J.Rumbaugh <i>rassemblés par Rational Software</i>	
1996	UML 0.9	G.Booch J.Rumbaugh I.Jacobson <i>rassemblés par Rational Software</i>	
Fin 1996		OMG	Appel d'offres pour une méthodologie
Début 1997	UML 1.0	G.Booch J.Rumbaugh I.Jacobson <i>rassemblés par Rational Software</i> ainsi que d'autres partenaires (Digital, HP, Microsoft...)	Soumission à l'OMG <i>Remarque : l'OMG a reçu une offre concurrente d'UML</i>
Fin 1997	UML 1.1	Les auteurs d'UML 1.0 et ceux de l'offre concurrente (dont Softeam)	Rapprochement et standardisation de la nouvelle version par l'OMG
Mars 1999	UML 1.3	Task Force constituée par l'OMG	Amélioration de la version 1.1
En cours	UML 2.0	Revision Task Force, constituée par l'OMG	Amélioration de la version 1.3

Figure 2.1 : Les stades de constitution d'UML

2.3 Les points forts et les points faibles D'UML :

2.3.1 Les points forts d'UML

- ❖ UML est un langage formel et normalisé :
 - gain de précision
 - gage de stabilité
 - encourage l'utilisation d'outils
- ❖ UML est un support de communication performant :
 - Il cadre l'analyse.
 - Il facilite la compréhension de représentations abstraites complexes.
 - Son caractère polyvalent et sa souplesse en font un langage universel.

2.3.2 Les points faibles d'UML

- ❖ La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

Même si l'Espéranto est une utopie, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

- ❖ Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

2.4 Les diagrammes d'UML :

Les 14 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

2.4.1 Les diagrammes de structurels ou statiques :

Les diagrammes structurels ou statiques (Structure Diagram) rassemblent :

- Diagramme de classes (Class diagram) : il représente les classes intervenant dans le système.

- Diagramme d'objets (Object diagram) : il sert à représenter les instances de classes (objets) utilisées dans le système.
- Diagramme de composants (Component diagram) : il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)
- Diagramme de déploiement (Deployment diagram) : il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
- Diagramme des paquetages (Package diagram) : un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions.
- Diagramme de structure composite (Composite Structure Diagram) : depuis UML 2.x, permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.
- Diagramme de profils (Profile diagram (en)) : depuis UML 2.2, permet de spécialiser, de personnaliser pour un domaine particulier un méta-modèle de référence d'UML.

2.4.2 Diagrammes comportementaux :

Les diagrammes comportementaux (Behavior Diagram) rassemblent

- Diagramme des cas d'utilisation (use-cases ou Use Case Diagram) : il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenant extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
- Diagramme états-transitions (State Machine Diagram) : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.
- Diagramme d'activité (Activity Diagram) : permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

2.4.3 Diagrammes d'interaction ou dynamiques :

Les diagrammes d'interaction ou dynamiques (Interaction Diagram) rassemblent :

- Diagramme de séquence (Sequence Diagram) : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

- Diagramme de communication (Communication Diagram) : depuis UML 2.x, représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- Diagramme global d'interaction (Interaction Overview Diagram) : depuis UML 2.x, permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
- Diagramme de temps (Timing Diagram) : depuis UML 2.3, permet de décrire les variations d'une donnée au cours du temps.

2.5 Les diagrammes de structurels ou statiques :

2.5.1 Diagramme de classes :

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

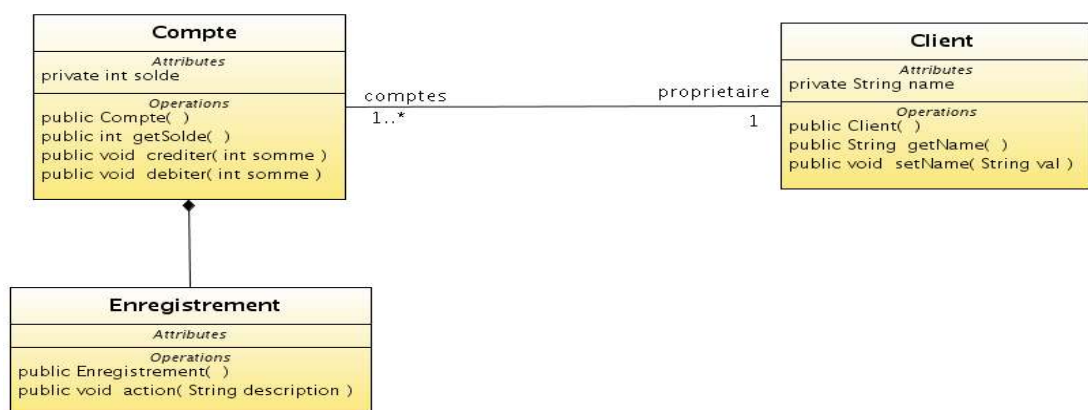


Figure 2.2 : exemple de diagramme de classe

2.5.2 Diagramme d'objets :

Dans le langage de modélisation de donnée UML, permet de représenter les instances des classes, c'est-à-dire des objets. Comme le diagramme de classes, il exprime les relations qui existent entre les objets, mais aussi l'état des objets, ce qui permet d'exprimer des contextes d'exécution. En ce sens, ce diagramme est moins général que le diagramme de classes.

Les diagrammes d'objets s'utilisent pour montrer l'état des instances d'objet avant et après une interaction, autrement dit c'est une photographie à un instant précis des attributs et objet existant. Il est utilisé en phase exploratoire.

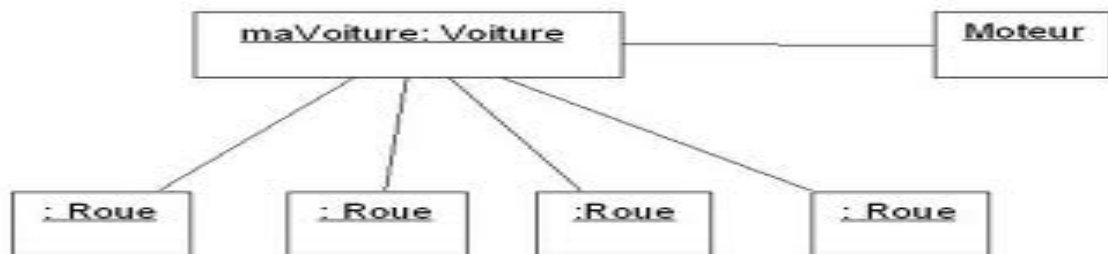


Figure 2.3 : exemple de diagramme d'objet

2.5.3 Diagramme de composants :

Décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les composants

2.5.4 Diagramme de déploiement :

Est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.

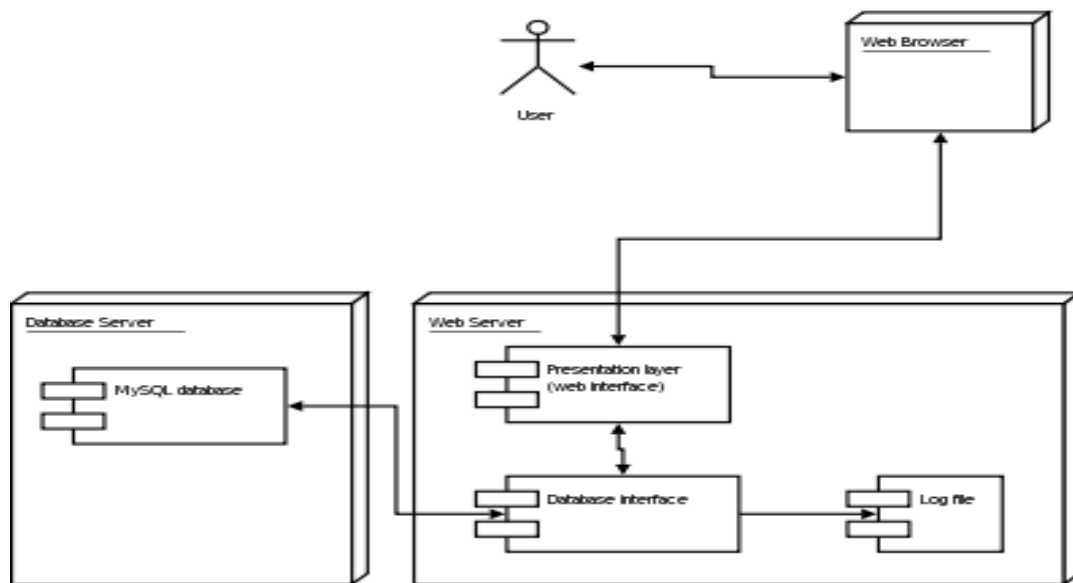


Figure 2.4 : exemple de diagramme de déploiement

2.6 Diagrammes comportementaux :

2.6.1 Diagramme des cas d'utilisation :

Sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. 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, et 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 peut seulement donner 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 liés, les cas d'utilisation sont bien plus détaillés que les diagrammes de cas d'utilisation.

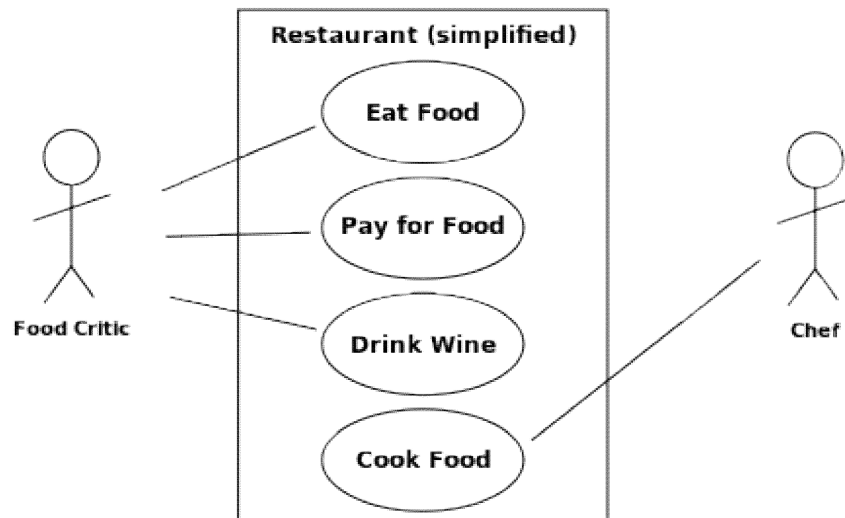


Figure 2.5 : exemple de diagramme de cas d'utilisation

2.6.2 Diagramme états-transitions :

Est un schéma utilisé en génie logiciel pour représenter des automates déterministes. Il fait partie du modèle UML et s'inspire principalement du formalisme des statecharts et rappelle les grafjets des automates. S'ils ne permettent pas de comprendre globalement le fonctionnement du système, ils sont directement transposables en algorithme. Tous les automates d'un système s'exécutent parallèlement et peuvent donc changer d'état de façon indépendante.

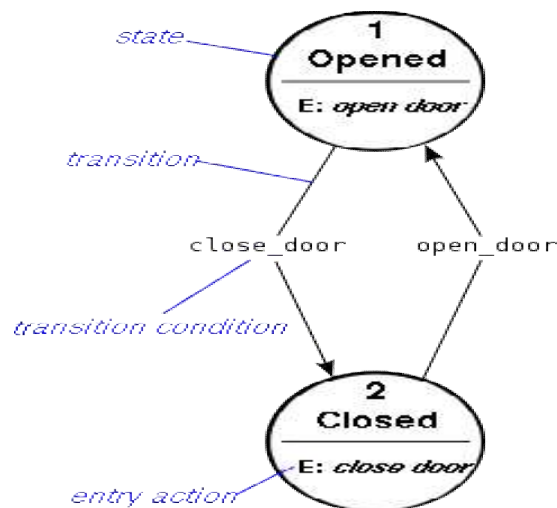


Figure 2.6 : exemple de diagramme d'état de transition

2.6.3 Diagramme d'activité :

est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multi-threads ou multi-processus). Le diagramme d'activité est également utilisé pour décrire un flux de travail (workflow).

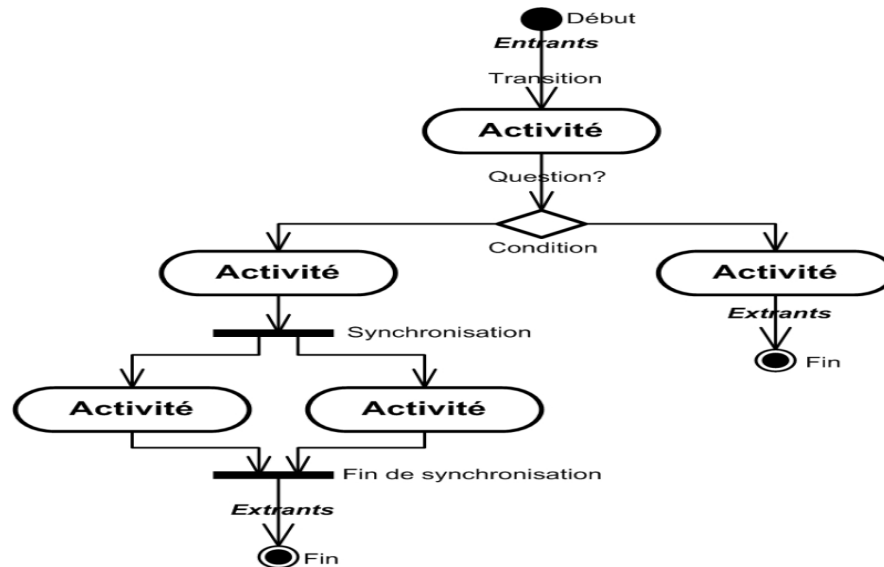


Figure 2.7 : exemple de diagramme d'activité

2.7 Diagrammes d'interaction ou dynamiques

2.7.1 Diagramme de séquence :

Sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

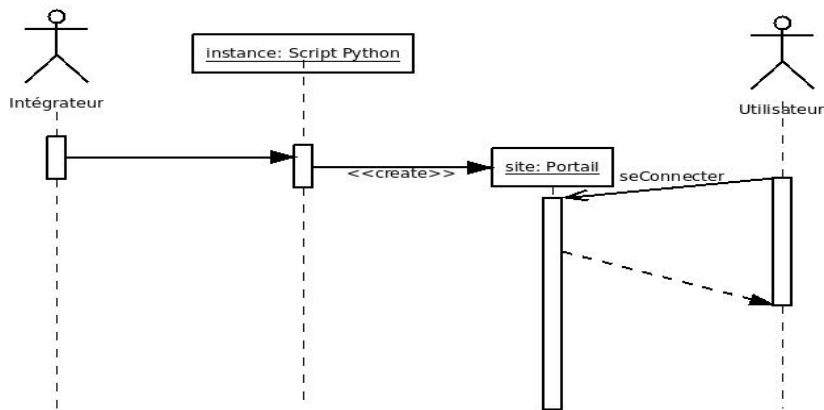


Figure 2.8 : exemple de diagramme de séquence

2.7.2 Diagramme de communication :

Est un diagramme d'interactions UML2.0 (appelé diagramme de collaboration en UML 1), représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets. En fait, le diagramme de séquence et le diagramme de communication sont deux vues différentes mais logiquement équivalentes (on peut construire l'une à partir de l'autre) d'une même chronologie, ils sont dits isomorphes.

C'est une combinaison entre le diagramme de classes, celui de séquence et celui des cas d'utilisation. Il rend compte à la fois de l'organisation des acteurs aux interactions et de la dynamique du système.

C'est un graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre objets.

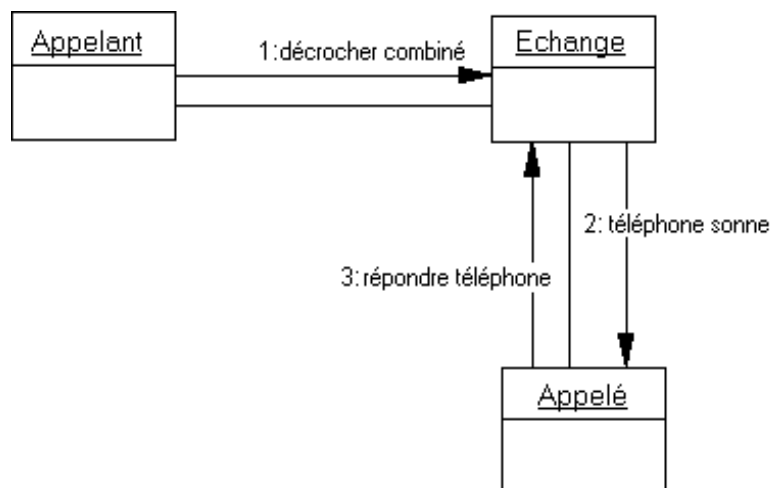


Figure 2.9 : diagramme de communication

3. Le processus unifié :

3.1 Définition:

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

3.2 Différent caractéristique du processus unifié :

3.2.1 UP est itératif :

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes. Elle qualifie un traitement ou une procédure qui exécute un groupe d'opérations de façon répétitive jusqu'à ce qu'une condition bien définie soit remplie.

Une itération prend en compte un certain nombre de cas d'utilisation et traite en priorité les risques majeurs.

3.2.2 UP est centré sur l'architecture :

Ph.Kruchten propose différentes perspectives, indépendantes et complémentaires, qui permettent de définir un modèle d'architecture (publication IEEE, 1995).

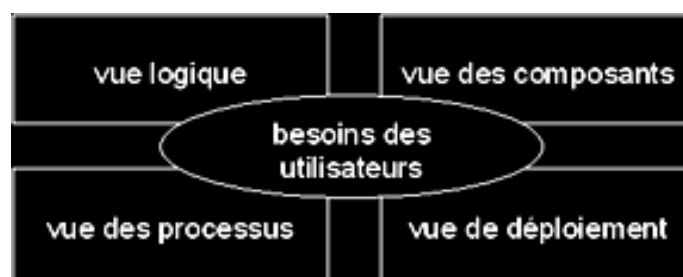


Figure 2.10 : La vue ('4+1')

- **La vue des cas d'utilisation** : contient les scénarios principaux qui sont utilisés pour faire fonctionner l'architecture et pour la valider.
- **La vue logique** : identifie la plupart des paquetages, sous-systèmes et classes.
- **La vue d'implémentation** : décrit l'organisation des modules du logiciel.
- **La vue des processus** : concerne les aspects concurrents du système à l'exécution: tâches, threads et leur interaction.
- **La vue de déploiement** : décrit les ressources matérielles et la répartition du logiciel dans ces ressources.

3.3 UP est piloté par les cas d'utilisation d'UML :

Le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc accès sur l'utilisateur.

Les cas d'utilisation permettent d'illustrer ces besoins.

Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.

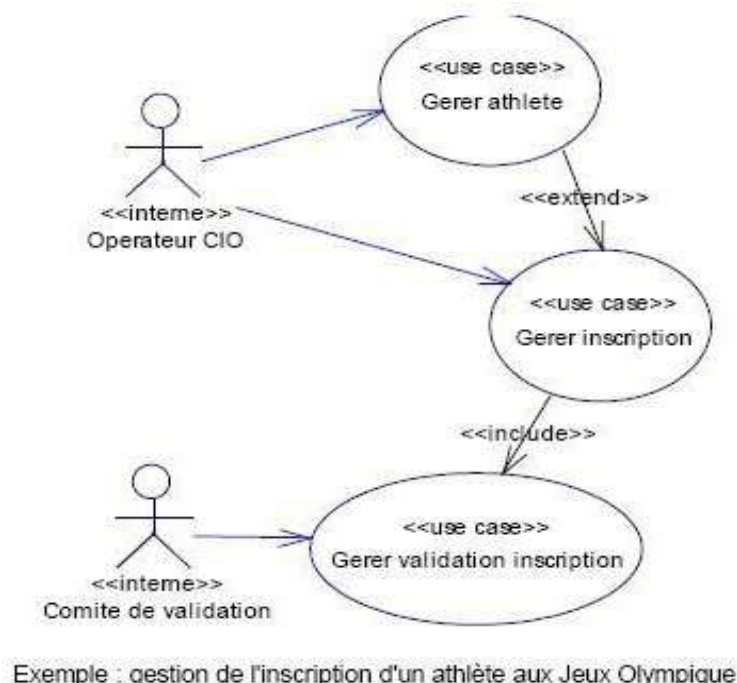


Figure 2.11 : cas d'utilisation UML pilote UP

3.4 Cycle vie du processus unifié :

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques.

UP est un ensemble de principes génériques adapté en fonctions des spécificités des projets. UP répond aux préoccupations suivantes :

- **QUI** participe au projet ?
- **QUOI**, qu'est-ce qui est produit durant le projet ?
- **COMMENT** doit-il être réalisé ?
- **QUAND** est réalisé chaque livrable ?

3.4 L'architecture bidirectionnelle :

UP gère le processus de développement par deux axes.

- **L'axe vertical** représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en terme de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.
- **L'axe horizontal** représente le temps et montre le déroulement du cycle de vie du processus; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons.

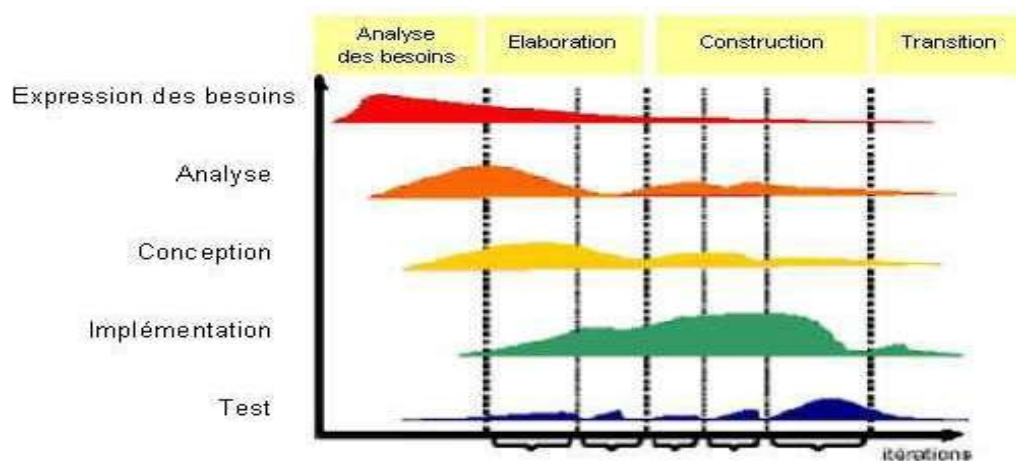


Figure 2.12 : présentation de cycle de vie processus

UP répète un certain nombre de fois une série de cycle qui s'articule autour de 4 phases

- analyse des besoins
- élaboration
- construction
- transition

Pour mener efficacement un tel cycle, les développeurs ont besoin de toutes les représentations du produit logiciel

- un modèle de cas d'utilisation
- un modèle d'analyse : détailler les cas d'utilisation et procéder à une première répartition du comportement
- un modèle de conception : finissant la structure statique du système sous forme de sous systèmes, de classes et interfaces.
- un modèle d'implémentation : intégrant les composants
- un modèle de déploiement : définissant les nœuds physiques des ordinateurs
- un modèle de test : décrivant les cas de test vérifiant les cas d'utilisation
- une représentation de l'architecture

3.5 Les activités du processus :

3.5.1 Expression des besoins :

L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

- inventorier les **besoins principaux** et fournir une liste de leurs fonctions
- recenser les **besoins fonctionnels** (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation
- appréhender les **besoins non fonctionnels** (technique) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

3.5.2 Analyse :

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

3.5.3 Conception :

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation.

Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.

Elle constitue un point de départ à l'implémentation :

- elle décompose le travail d'implémentation en sous-système
- elle crée une abstraction transparente de l'implémentation

3.5.4 Implémentation :

L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

3.5.5 Test :

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction.

Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

3.6 Les phases :

3.6.1 Analyse des besoins :

L'analyse des besoins donne une vue du projet sous forme de produit fini.

Cette phase porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur),

l'architecture générale du système, les risques majeurs, les délais et les coûts
On met en place le projet.

Elle répond aux questions suivantes :

- que va faire le système ? par rapport aux utilisateurs principaux, quels services va-t-il rendre?
- quelle va être l'architecture générale (cible) de ce système
- quels vont être : les délais, les coûts, les ressources, les moyens à déployer?

3.6.2 Elaboration :

L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en oeuvre.

L'élaboration permet de préciser la plupart des cas d'utilisation, de concevoir l'architecture du système et surtout de déterminer l'architecture de référence.

Au terme de cette phase, les chefs de projet doivent être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.

Les tâches à effectuer dans la phase élaboration sont les suivantes :

- créer une architecture de référence
- identifier les risques, ceux qui sont de nature à bouleverser le plan, le coût et le calendrier
- définir les niveaux de qualité à atteindre
- formuler les cas d'utilisation pour couvrir les besoins fonctionnels et planifier la phase de construction
- élaborer une offre abordant les questions de calendrier, de personnel et de budget

3.6.3 Construction :

La construction est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet.

Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

3.6.4 Transition :

Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts.

Cette phase suppose des activités comme la formation des utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées.

CHAPITRE 3

1. introduction :

Le but du site création site web dynamique pour la faculté science et technologie, doit être impérativement procéder d'une méthodologie de conception permettant d'offrir une vision claire des différents éléments de l'application et de l'interaction entre ces derniers

Dans ce chapitre, nous allons donc présenter les différentes étapes conceptuelles de notre site web, tout en suivant la démarche du processus UP afin d'aboutir à une implémentation physique englobant les différents aspects d'un site (interface utilisateur, le parcours, ...etc.) Pour la conception de notre projet, nous allons suivre le processus de développement UP, avec une modélisation UML.

2. Exigences fonctionnelles :

Notre projet porte sur la conception et la réalisation d'un site web dynamique pour la faculté science et technologie, ce site est destiné aux étudiants ce site peut regrouper les fonctionnalités suivantes :

a) Authentification :

Ce service permet à tous les membres titulaires sur ce site, de s'identifier pour accéder à leur espace privé et aussi aux différents services privés destinés qu'à eux seulement.

b) ajouter les emplois des temps :

Le site nous permet d'ajouter les emplois des temps Pour différents niveaux de ce par Admin Et l'étudiant peut le voir

c) ajouter les cours :

Le professeur ajoute des cours afin de permettre à l'étudiant de rester sur l'étude toujours connecté à distance

d) ajouter les départements :

e) Ajouter professeurs

f) Ajouter étudiants

g) Télécharger les leçons

h) Afficher preuves League

i) Parcourir les emplois des temps

j) Parcourir nouveau à l'université

3. spécification des exigences d'après les cas d'utilisations :

3.1 identification des acteurs :

- a) Etudiant :
- b) Enseignant :
- c) Admine :

3.2 Identification des cas des d'utilisation :

➤ Admine

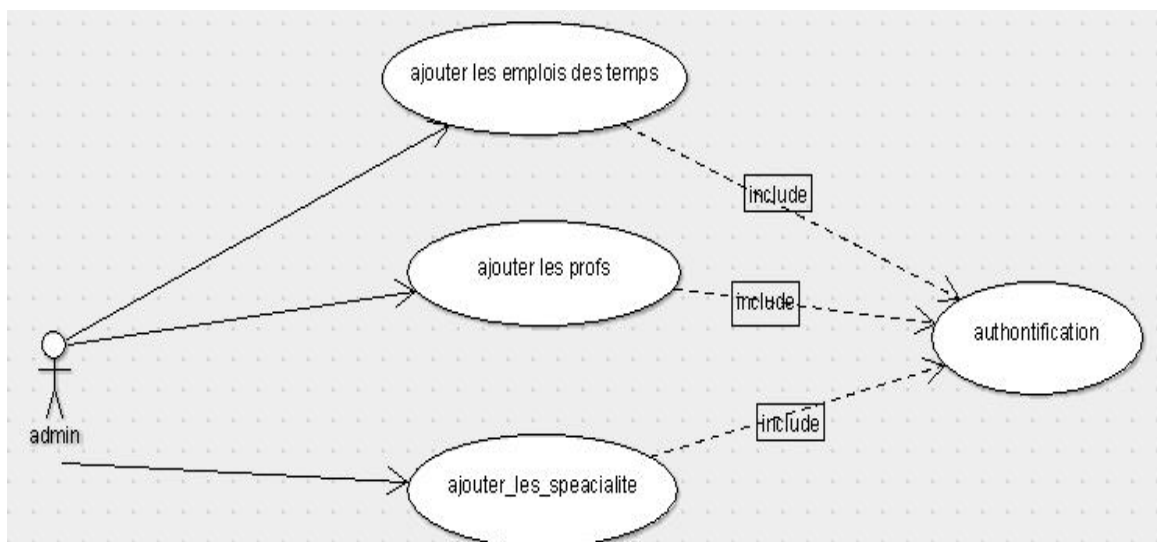


Figure 3.1 : cas d'utilisation pour l'admine

➤ **Etudiant :**

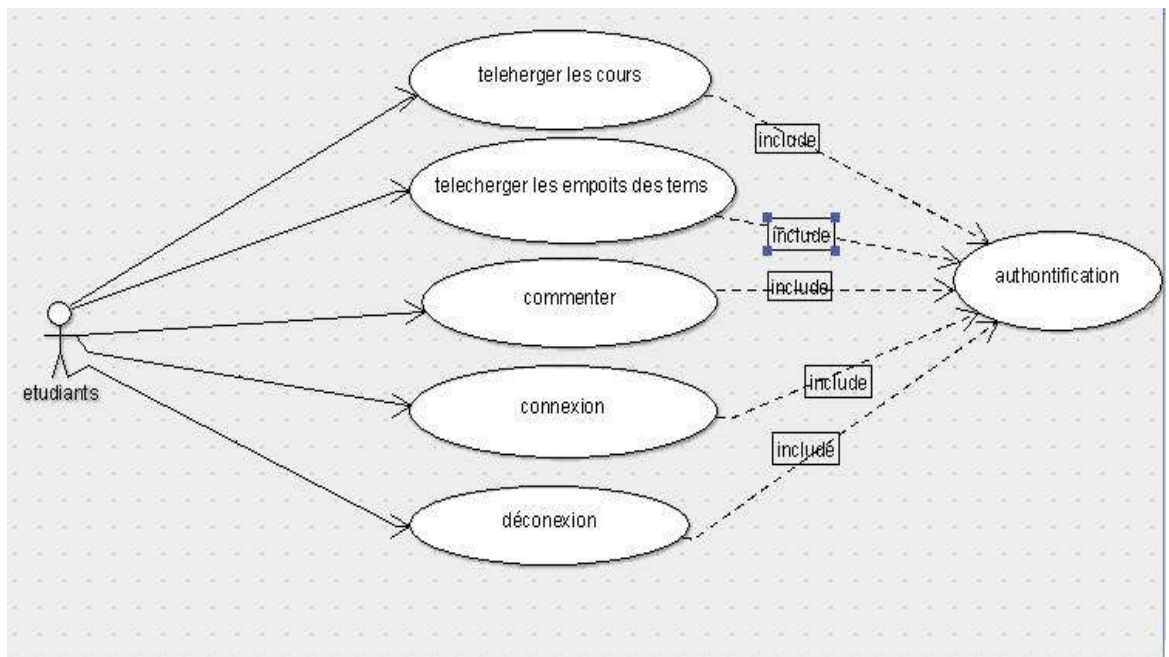


Figure 3.2 : cas d'utilisation pour Etudiant

➤ **Enseignant :**

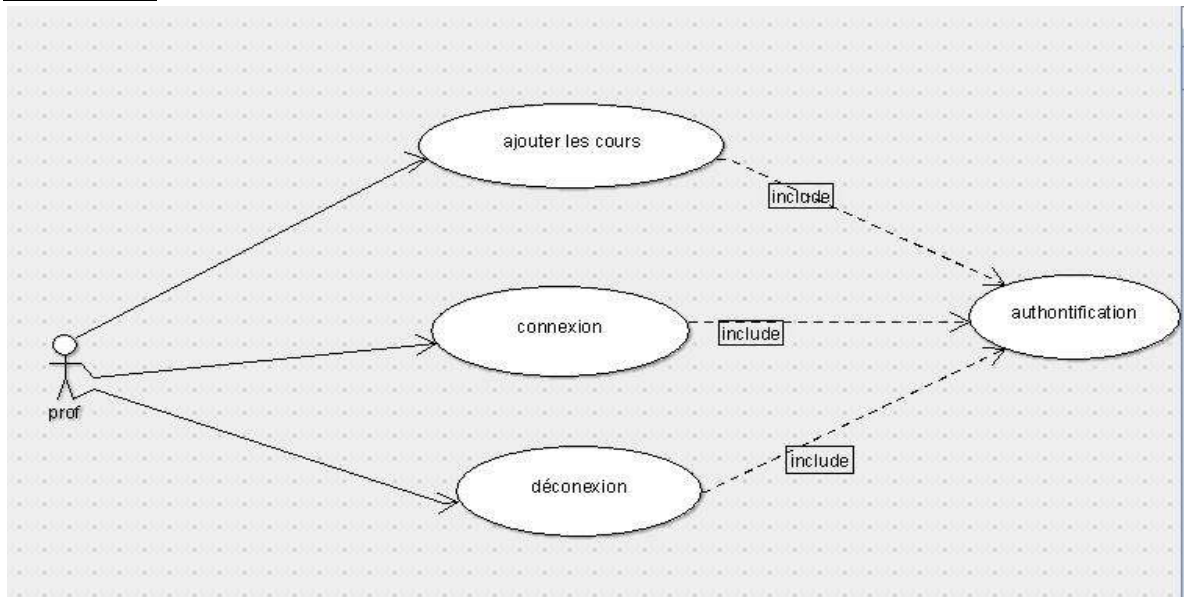


Figure 3.3 : cas d'utilisation pour Enseignant

4. Identification des concepts du domaine :

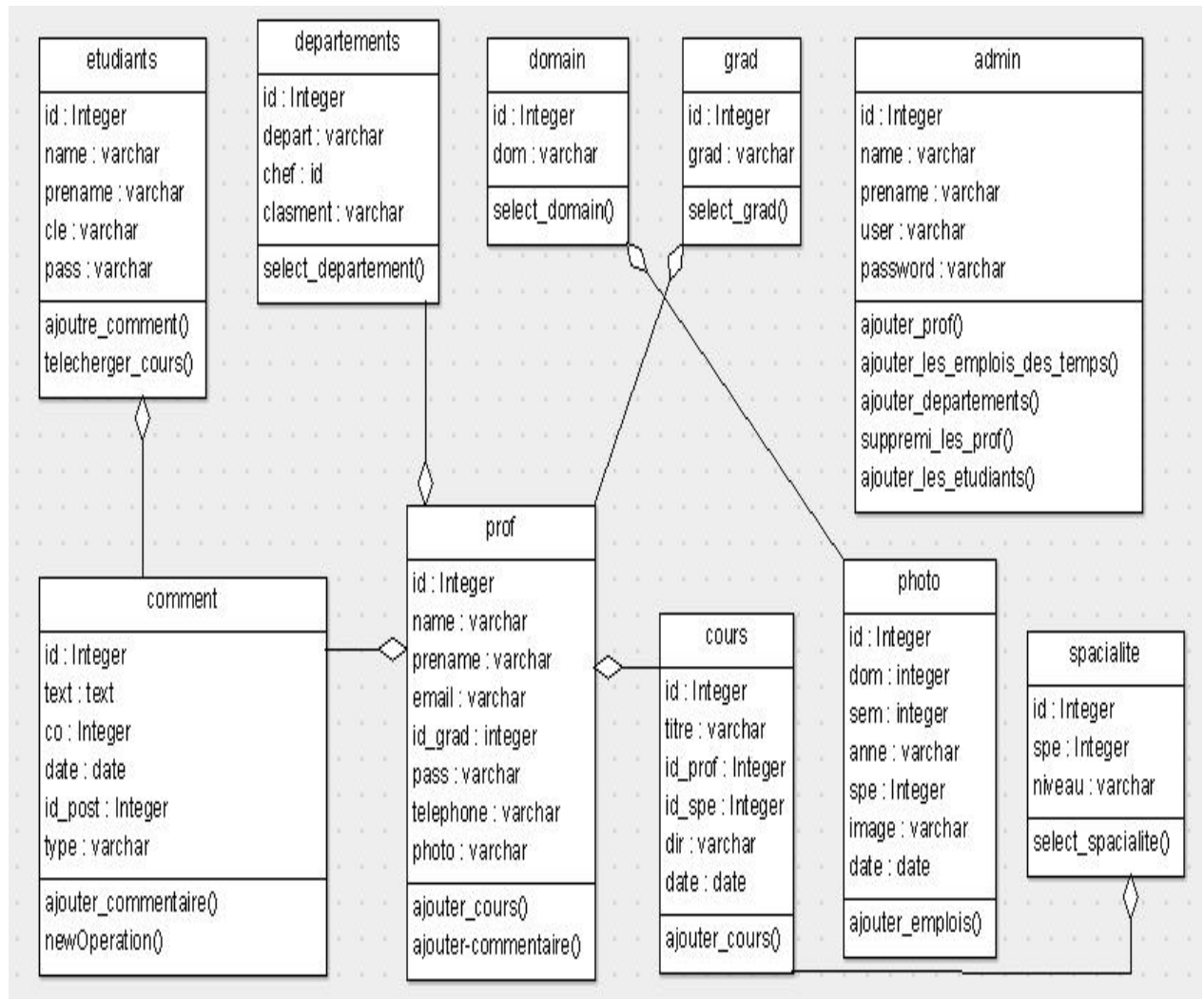


Figure 3.4 : Diagramme de classe du domaine

5. Diagrammes de séquence système :

a) Authentification : étudiant

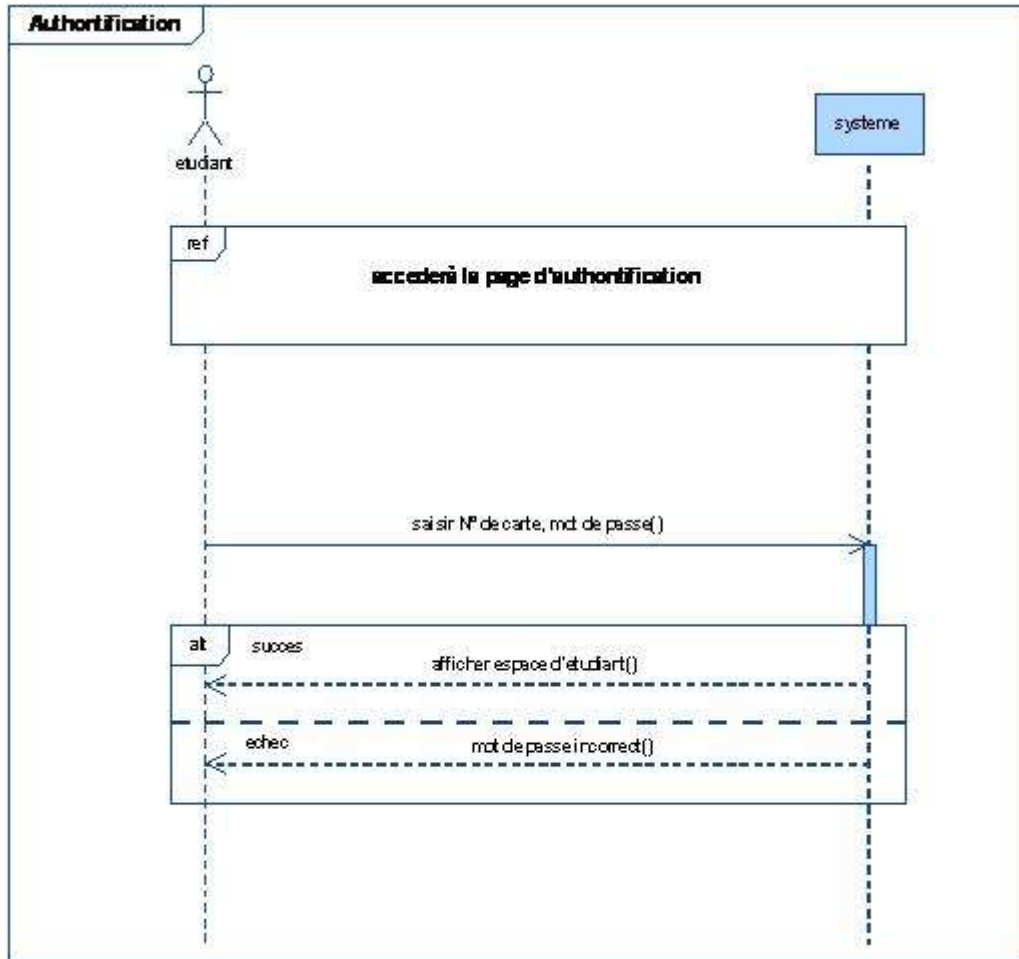


Figure 3.5 : diagramme de séquence connexion d'étudiants

b) **Authentification** : admin

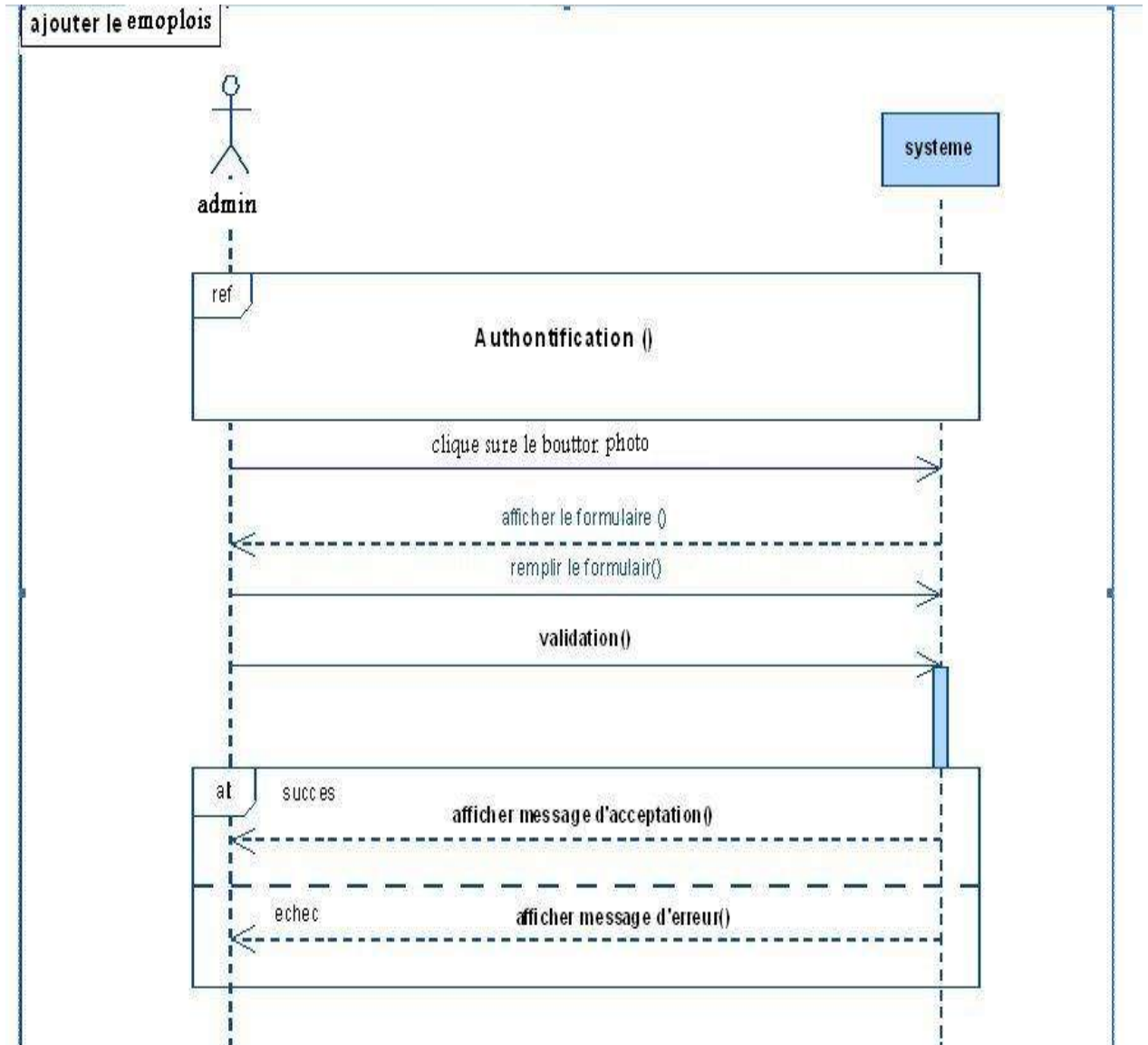


Figure 3.6 : diagramme de sequence ajouter les emplois des temps

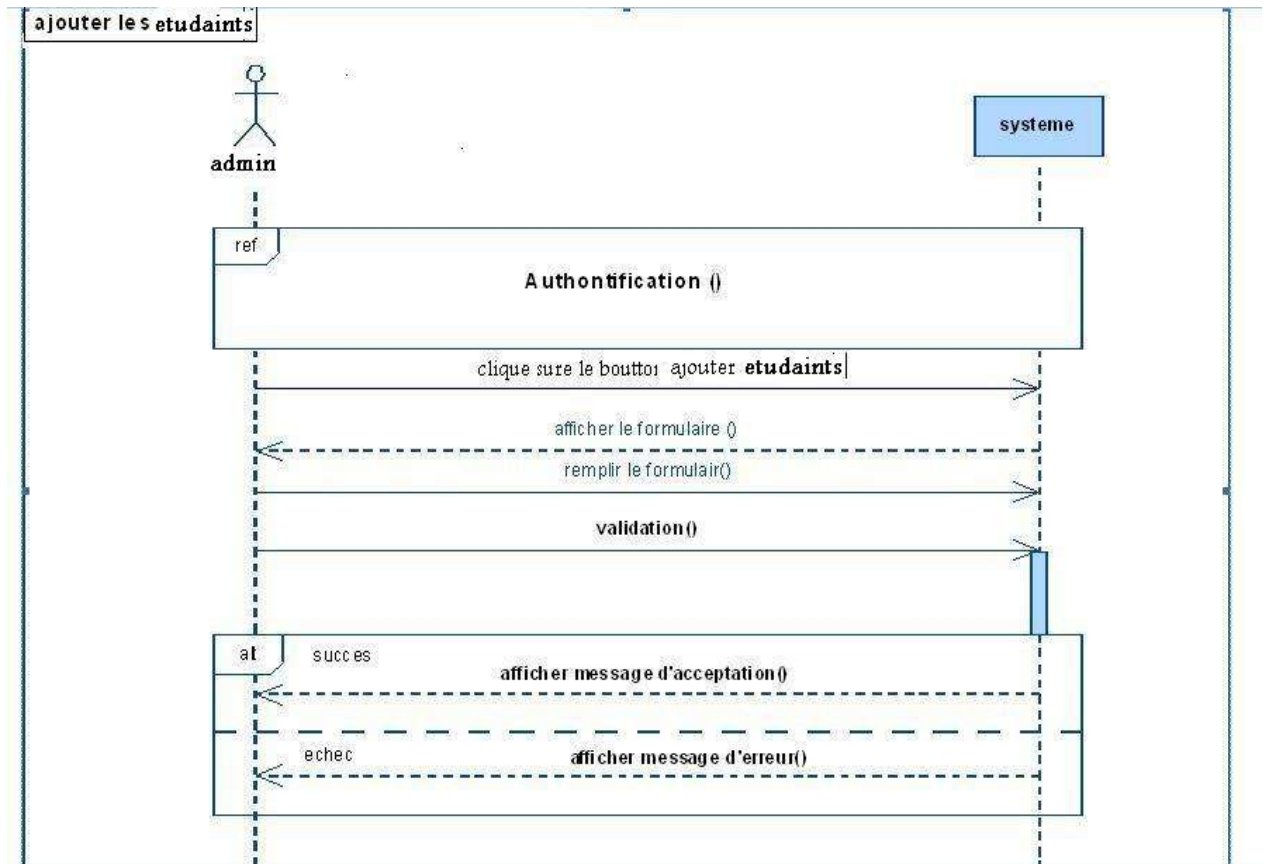


Figure 3.7: diagramme de sequence ajouter les etudiants

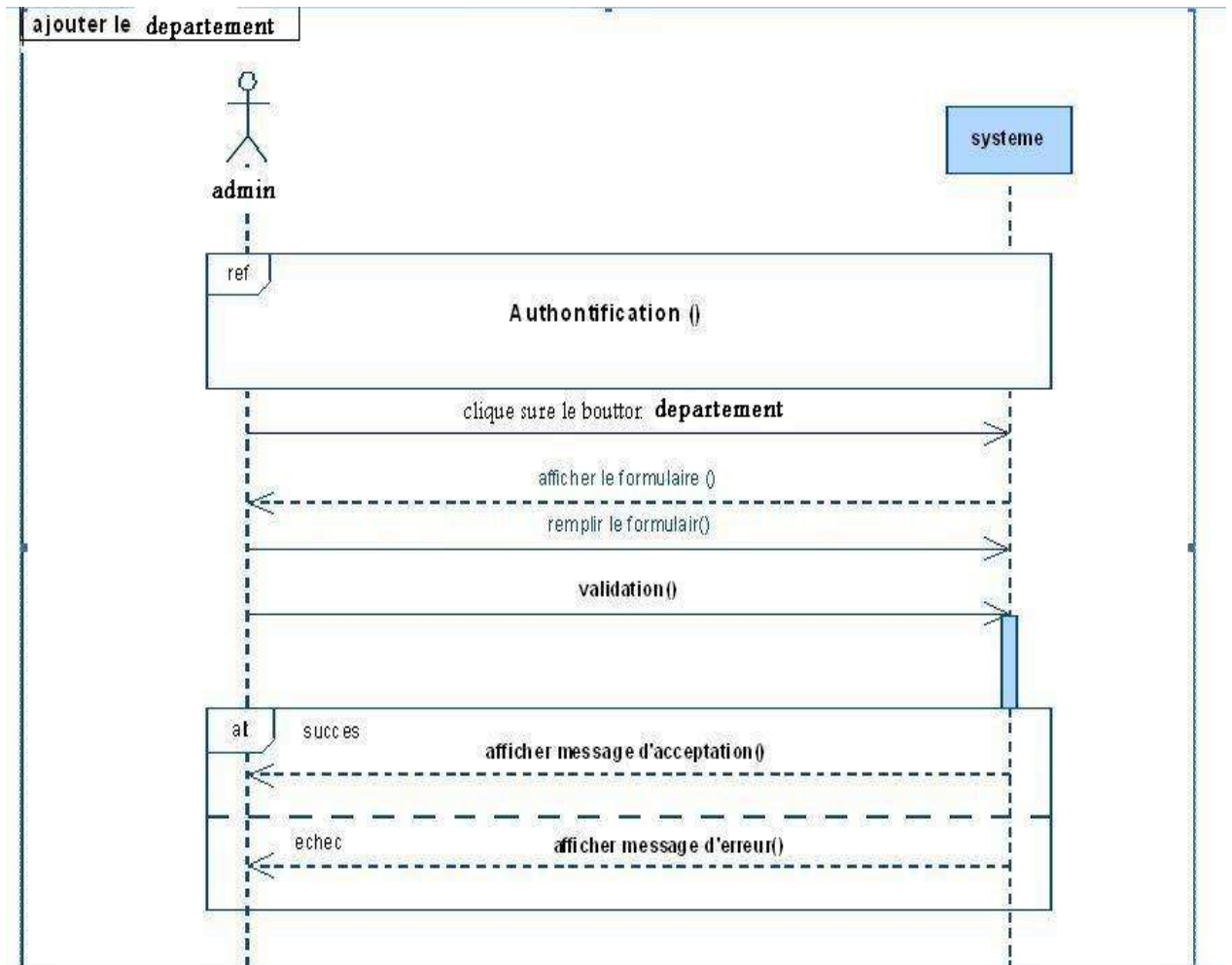


Figure 3.8: diagramme de sequence d'ajouter departements.

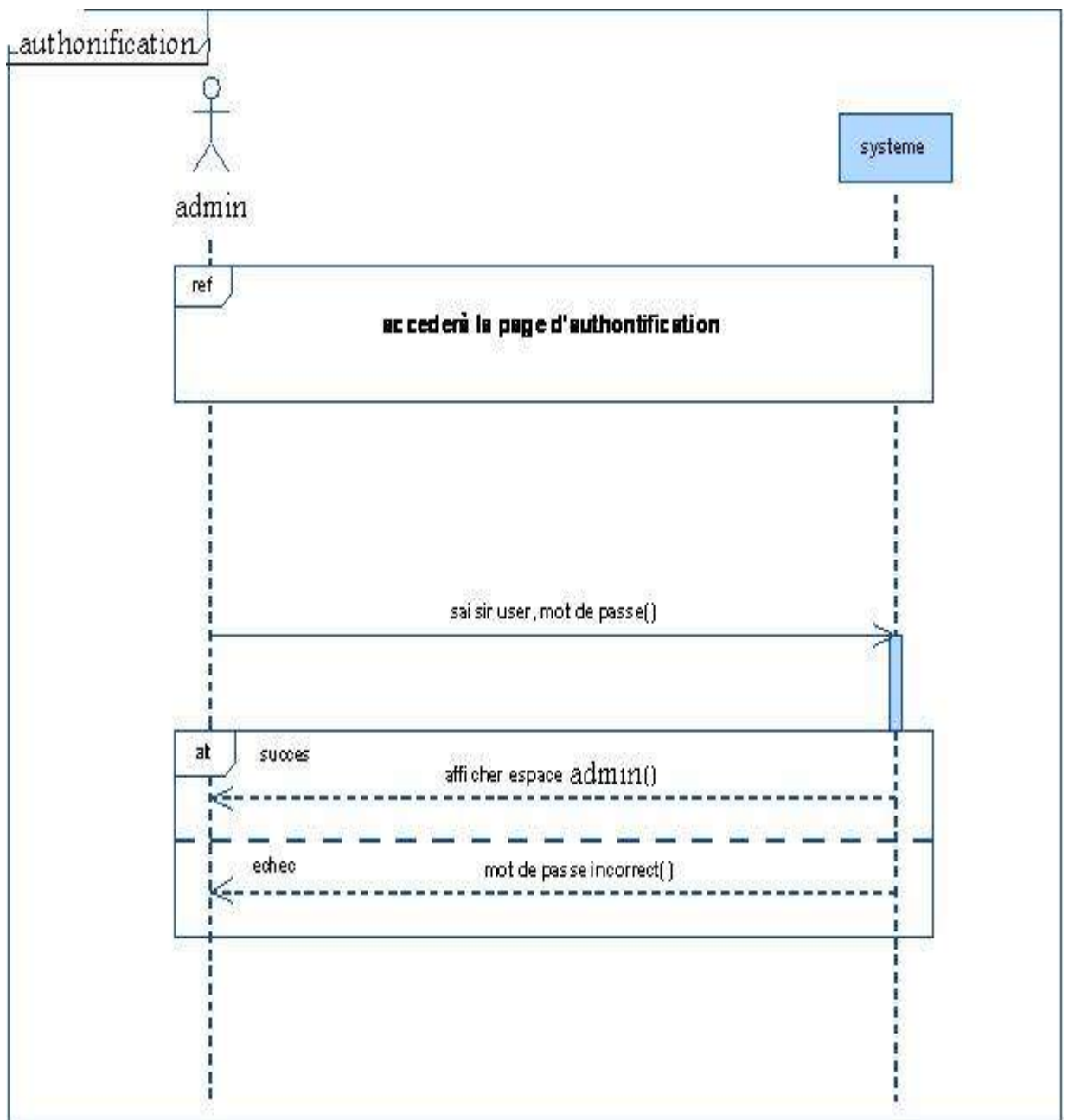


Figure 3.9 : diagramme de sequence connexion d'admin

c) **Authentification** : enseignant

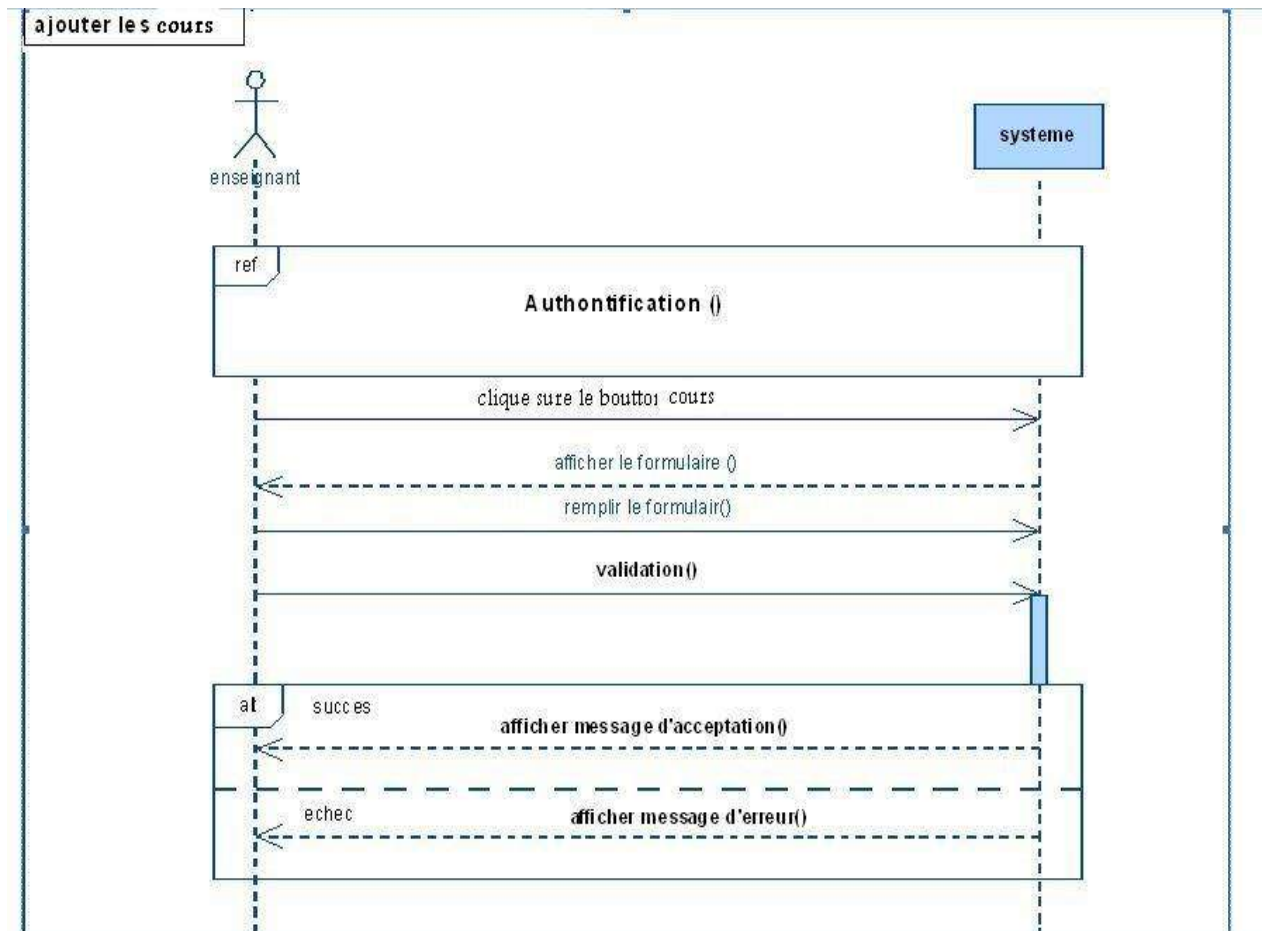


Figure 3.10 : diagramme de sequence ajouter les cours.

6. Conclusion :

Ce chapitre été consacré à l'analyse et la conception de notre système. Après avoir passé par les étapes du processus du développement UP nous faisons aux diagrammes d'UML pour les applications web, on a pu faire une modélisation basé UML englobons la partie de conception de l'application.

Le chapitre suivant sera réservé à la partie de réalisation du site, on va vous présentez une spécification de la configuration matérielle des logicielle des machines sur lequel on à testé l'application, ainsi qu'une présentation des logiciels utilisés, en motivons le choix ces outils à savoir le langage de programmation, et l'éditeur de pages web...etc.

CHAPITRE 4

1. Introduction :

Après avoir analysé des besoins et définie la méthodologie de conception, nous allons programmer l'essentiel de notre conception via un environnement adéquat.

Dans ce chapitre, nous allons décrire les logiciels utilisés, nous présenterons quelque exemples de code ainsi que des interfaces utilisateur représentant des pages web qui on été réalisées

2. environnement de développement :

Puisque cette application est un site dynamique nécessitant la disponibilité d'une infrastructure serveur, plusieurs applications sont nécessaires du coté serveur pour le faire fonctionner :

- Un serveur web (le serveur APACHE).
- Un langage de script serveur (PHP).
- Un serveur de base de données (MYSQL).
- Rapide PHP 2014 qui est considère comme des éditeurs HTML.

2.1 Le langage de programmation :

2.1.1 PHP (Hypertext Preprocessor) :

PHP est un langage de programmation informatique essentiellement utilisé pour produire à la volée des pages web dynamiques et la création d'applications dynamiques, le plus souvent dédiées au web. PHP peut être installé sur les principaux serveurs web du marché, les plus répandus étant IIS et Apache. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.



Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de réseau une bonne compréhension des mécanismes sous-jacents ainsi qu'une connaissance des problèmes de sécurité

2.1.2 MYSQL (structure query langage) :

- ✓ MySQL est un serveur de bases de données relationnelles stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" est: le langage standard pour les traitements de bases de données.
- ✓ MySQL, est un système de gestion de base de données relationnelle. Il est basé sur la langue de la structure d'interrogation ([SQL](#)), qui est utilisé pour l'ajout, la suppression et la modification des informations dans la base de données. Commandes SQL standard, telles qu'ADD, DROP, INSERT et UPDATE peuvent être utilisés avec MySQL



➤ Connexion à la base de données MYSQL :

Pour pouvoir vous connecter depuis une page PHP à votre base de données MySQL, il faudra spécifier plusieurs paramètres :

- * l'hôte (le serveur sur lequel MySQL est installé).
- * le login utilisateur.
- * le mot de passe.
- * le nom de la base de données.

Par défaut, les paramètres mis en place par APPSERV sont :

- * hôte ou serveur : "localhost".
- * username ou login : "root".
- * mot de passe : "".

La connexion au serveur MySQL s'effectue par la fonction `mysql_connect()`. Sa syntaxe est :

```
$mysqli_connect= mysql_connect('mysql_host', 'mysql_user', 'mysql_password', 'basse_donnee')
```

2.1.3Le serveur apache :

Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les [systèmes d'exploitation](#) de type [Unix](#), mais il a désormais été porté sur de nombreux [systèmes](#), dont [Microsoft Windows](#).

2.1.4 Appserv :

Une solution simple et efficace pour installer sur un poste Windows l'ensemble des composants Apache, PHP, MySQL et Perl.

Ceci permet de disposer d'un poste autonome pour effectuer le développement et les tests d'applications Web.

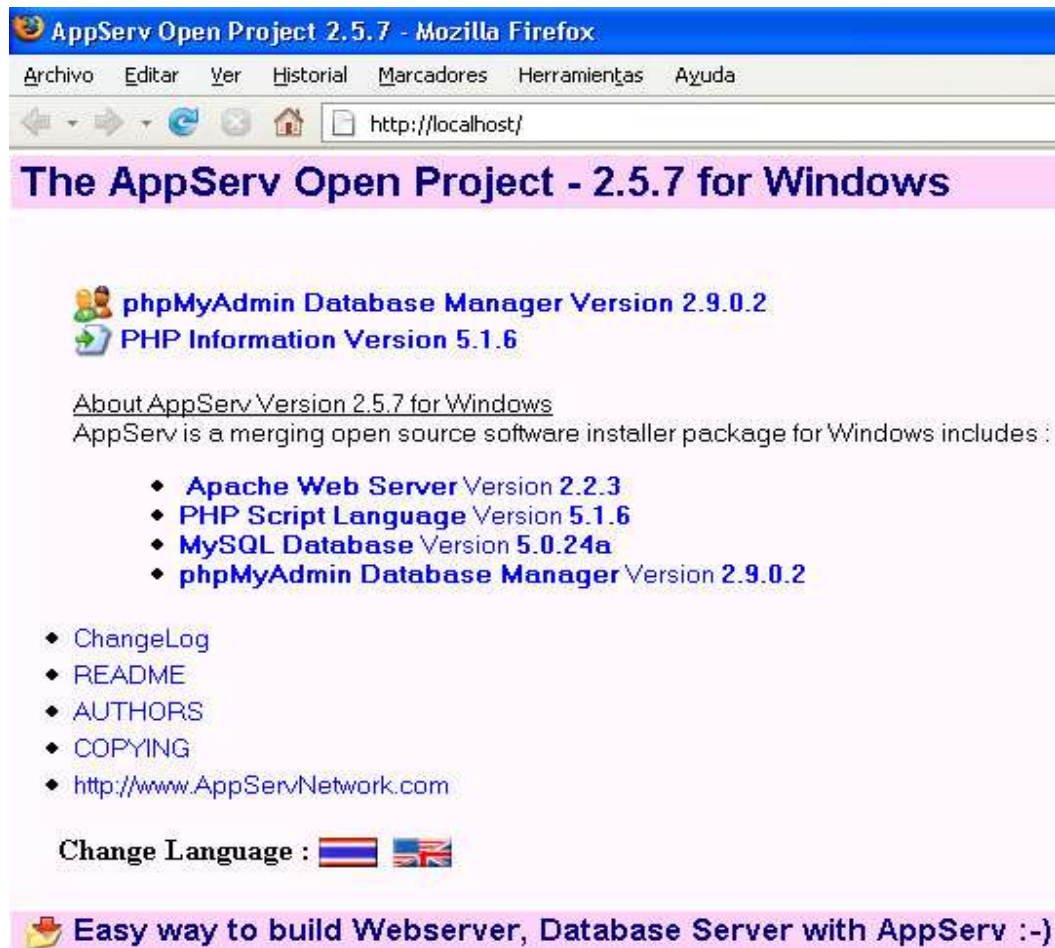


Figure 4.1 : interface d'appserv

2.1.5 PHP My Admin:

phpMyAdmin est un outil logiciel libre écrit en [PHP](#), destiné à gérer l'administration de [MySQL](#) sur le Web. phpMyAdmin gère un large éventail d'opérations sur MySQL, Opérations utilisées fréquemment (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des permissions, etc) peuvent être effectuées par l'intermédiaire de l'interface utilisateur, alors que vous avez encore la possibilité d'exécuter directement une instruction SQL

2.1.7 La base de données :

Une base de données (son abréviation est BD, en anglais DB, data base) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des

programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de [réseau](#), afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

2.1.8 Langage HTML (hyper texte markup language) :

HTML est un langage de description de document utilisé sur Internet pour faire des pages Web. Son sigle signifie « HyperText Markup Language » en [anglais](#), littéralement « langage de marquage hypertexte ». Le balisage HTML est incorporé dans le texte du document et est interprété par un navigateur Web.

Le langage HTML permet notamment la lecture de documents sur [Internet](#) à partir de machines différentes, grâce au [protocole HTTP](#), permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée [URL](#).

2.1.9 CSS (Cascading Style Sheets) :

Littéralement Cascading Style Sheets (feuilles de style ne cascade), CSS est un langage déclaratif simple pour mettre en forme des pages HTML ou des documents XML. Le langage CSS permet

de préciser les caractéristiques visuelles et sonores de présentation d'une page Web : les polices de caractères, les marges et bordures, les couleurs, le positionnement des différents éléments, etc. Le terme de "Cascading" Style Sheets sous entend qu'il est possible de définir un style pour une page HTML puis, à l'intérieur de cette même page, de fournir des informations plus précises ou différentes pour présenter certains éléments plus distinctement.

Exemple:

```
<HTML>
<HEAD>
  <STYLE type="text/css">
    <!-- Définition des styles; -->
  </STYLE>
</HEAD>
</HTML>
```

2.1.10 JavaScript :

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,

Le Javascript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du [serveur](#) web.



2.2 Les outils :

2.2.1 Rapide PHP 2014 :

Rapide éditeur PHP est un éditeur PHP rapide et plus puissant pour Windows combinant les caractéristiques d'un IDE PHP entièrement emballé avec la vitesse du Bloc-notes. Rapide PHP est le logiciel le plus complet tout-en-un pour le codage PHP, HTML, CSS, JavaScript et d'autres

langages de développement web avec des outils pour le débogage, la validation, la réutilisation, la navigation et le formatage de votre code. Avec Rapid PHP Editor vous pouvez coder plus intelligent, de gagner du temps et augmenter la productivité.

3. Présentation de l'application :

3.1 Conception et réalisation de la base de données :

Notre base de données s'appelle : **khenchela** , elle comporte 10 tables, chaque table à réservée pour une fonctionnalité de notre application web, par exemple la table **_etudiants** contient les informations de les étudiants, la table **grad** contient les grads ...etc, on va vous présentez une description de notre base de données.



The screenshot shows the phpMyAdmin interface for a MySQL database named 'khenchela' on localhost. The interface includes a sidebar with navigation icons and a list of tables. The main area displays a table structure view with columns for Table, Action, Enregistrements, Type, Interclassement, Taille, and Perte. The tables listed are: admin, comment, cours, departements, domain, etudiants, grad, photo, prof, and spacialite. A summary row at the bottom indicates 10 tables with a total of 39 records and a size of 20.3 Kio.

Table	Action	Enregistrements	Type	Interclassement	Taille	Perte
<input type="checkbox"/> admin	[Icons]	1	MyISAM	utf8_general_ci	2,0 Kio	-
<input type="checkbox"/> comment	[Icons]	0	MyISAM	utf8_general_ci	1,0 Kio	-
<input type="checkbox"/> cours	[Icons]	7	MyISAM	utf8_general_ci	2,6 Kio	36 o
<input type="checkbox"/> departements	[Icons]	0	MyISAM	utf8_general_ci	1,0 Kio	-
<input type="checkbox"/> domain	[Icons]	4	MyISAM	utf8_general_ci	2,1 Kio	-
<input type="checkbox"/> etudiants	[Icons]	2	MyISAM	utf8_general_ci	2,0 Kio	-
<input type="checkbox"/> grad	[Icons]	6	MyISAM	utf8_general_ci	2,2 Kio	-
<input type="checkbox"/> photo	[Icons]	2	MyISAM	utf8_general_ci	2,2 Kio	-
<input type="checkbox"/> prof	[Icons]	7	MyISAM	utf8_general_ci	2,7 Kio	236 o
<input type="checkbox"/> spacialite	[Icons]	10	MyISAM	utf8_general_ci	2,3 Kio	-
10 table(s)	Somme	39	MyISAM	utf8_general_ci	20,3 Kio	272 o

Figure 4.2 : Présentation de la base de données

4. implémentation et réalisation de l'application web :

4.1 Mise en place des pages du site :

Nous mettrons tous les éléments du site (fichiers PHP, HTML, CSS, images,.....etc) dans un répertoire, qui s'appelle faculté_st



Figure 4.3 : dossier de site

4.3 Arborescence du site :

L'arborescence des fichiers de notre site est organisée comme suite :

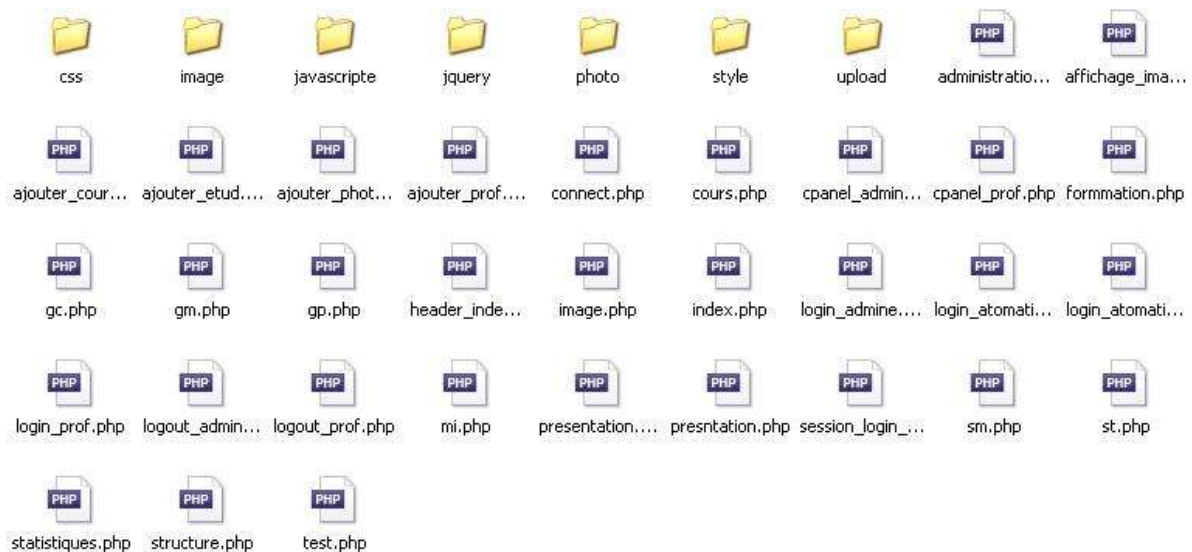


Figure 4.4 : Arborescence du site

5. Captures

5.1 Page d'accueil :

Chapitre 4: [IMPLEMENTATION]

The screenshot shows a web browser window displaying the website for the Faculty of Sciences and Technology at Université Abbas Laghrour Khenchela. The page is titled 'les cours' and features a navigation menu with options: Accueil, Presentation, Administration, Formations, Statistiques, and Structure. On the left, there is a 'Départements' sidebar listing: Mathématiques et Informatique, Sciences et de la technologie, Génie Mécanique, Génie Civil, Génie procédés, and Sciences de la Matière. The main content area displays two course listings:

- Course 1:** les cours de enseignant azou hoggas pour 1er licence math et informatique. titre : 2. telecharger : [download icon]. date : Thu-Apr-2015 00:44:44.
- Course 2:** les cours de enseignant azou hoggas pour 2eme mastre math. titre : 2. telecharger : [download icon]. date : Thu-Apr-2015 00:47:08.

On the right side, there is a 'Doyen de la Faculté' section with a photo of a man and the text 'Professeur : Zedira hamma'. Below this are buttons for 'VOTRE COMPTE' and 'Liens Utiles'. At the bottom left, a small text reads 'Transfert des données depuis www.comparinfoenrie.com...'.

Figure 4.5 : capture page d'accueil

5.2 Page connexion admin ou enseignant :

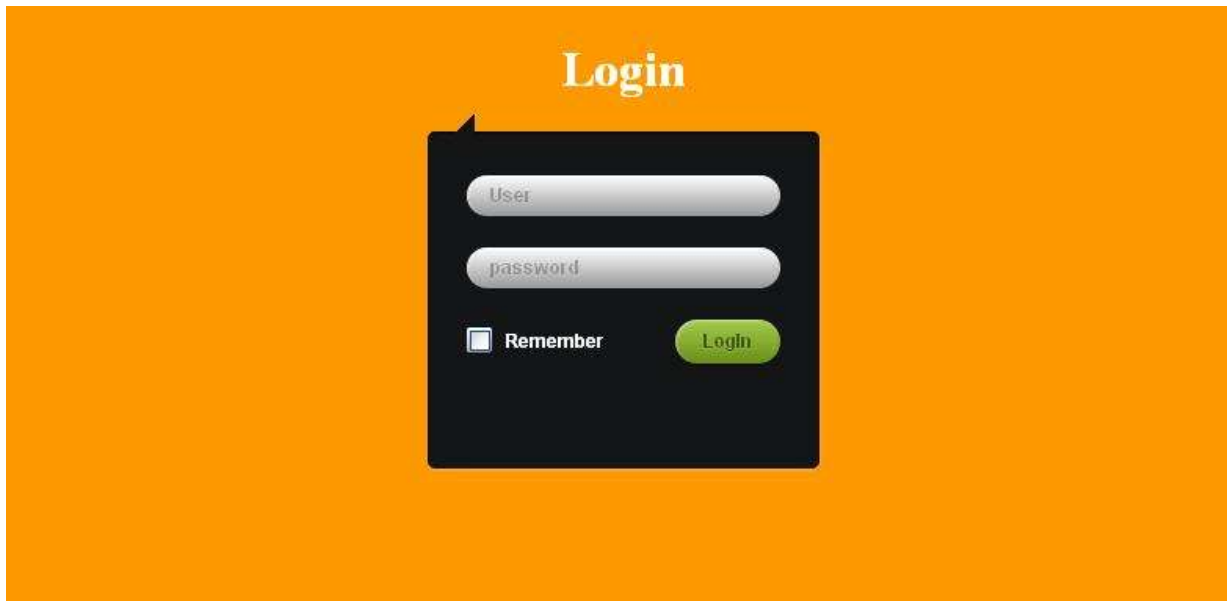


Figure 4.6 : Capture de Page adminne ou ensingant

5.3 page de profile adminne :



Figure 4.7 : Figure 4.3capture page de profile adminne

5.4 formulaire ajouter des ensingants :

The screenshot shows a web application interface titled "Cotrol Panel" (sic). At the top, there are three buttons: "Ajoute Prof", "Ajoute etud", and "Ajoute Prof" (highlighted in orange). To the right is a "logout" button. The main content area is a green box containing a form with the following fields:

- Name :
- Prenome :
- Email :
- Grad :
- Telephone :
- Password :

At the bottom of the form is an orange button labeled "ajouter".

Figure 4.8 : capture formulaire ajouter des enseignants

6. conclusion :

Ce chapitre a été consacré à la présentation des techniques de traitement utilisées pour la réalisation de notre projet, les nouvelles technologies, telles que le langage PHP pour notre page du côté serveur, et le JavaScript pour le traitement des pages du côté client, sans oublier le serveur pour la création et la gestion de notre base de données et le logiciel rapide PHP et l'éditeur Notepad++ pour la création des pages de site.

Conclusion générale

On retiendra de ce projet qu'une étude de faisabilité est indispensable avant le développement d'un site Internet. En effet, la création d'un site Internet n'est pas une activité à prendre à la légère, elle mérite d'être suivie et encadrée.

Les quatre mois passés à travailler, m'ont permis de réaliser que le projet d'un site Internet est la somme de plusieurs actions (ou étapes) planifiées et dépendantes les unes des autres.

Toutes les étapes de ce projet m'ont permis d'enrichir mon expérience. On retiendra également qu'un projet de site Internet demande une bonne organisation et une cohérence entre les différents acteurs du projet.

Les étapes du projet sont presque toutes dépendantes les unes des autres. Avant de passer à l'étape suivante, il faut, avant tout, valider l'étape précédente. Ce qui consomme du temps pour un projet est de revenir sur une étape après validation. Il est donc important de faire valider chaque étape du projet lors des séances de travail et en présence de tous les membres du groupe de travail. Toutefois cela ne doit pas empêcher les acteurs du projet de faire preuve d'ouverture d'esprit.

Pour la conduite d'un projet web, comme pour tous les autres projets, on est souvent obligé de faire des changements de dernière minute. La modification des choix initiaux peut intervenir à n'importe quel moment du projet et peut influencer les délais.

Pour conclure, il est important de préciser que la méthode proposée dans ce mémoire est le fruit d'une expérience durant laquelle j'ai moi-même eu l'occasion d'apprendre énormément.

Le but est de partager cette expérience en espérant qu'elle pourra inspirer les personnes qui souhaitent s'informer sur la conception de sites Internet.

TABLE DE FIGURES

CHAPITRE 1

FIGURE 1.1 : LES SITES STATIQUES.....02

FIGURE 1.2 : LES SITES DYNAMIQUE.....03

FIGURE 1.3 : CYCLE DE VIE D'UN SITE WEB03

FIGURE 1.4 : SCHEMA DE SITE WEB05

CHAPITRE 2

FIGURE 2.1 : LES STADES DE CONSTITUTION D'UML.....09

FIGURE 2.2 : EXEMPLE DE DIAGRAMME DE CLASSE12

FIGURE 2.3 : EXEMPLE DE DIAGRAMME D'OBJET13

FIGURE 2.4 : EXEMPLE DE DIAGRAMME DE DÉPLOIEMENT14

FIGURE 2.5 : EXEMPLE DE DIAGRAMME DE CAS D'UTILISATION15

FIGURE 2.6 : EXEMPLE DE DIAGRAMME D'ÉTAT DE TRANSITION 15

FIGURE 2.7 : EXEMPLE DE DIAGRAMME D'ACTIVITÉ16

FIGURE 2.8 : EXEMPLE DE DIAGRAMME DE SÉQUENCE17

FIGURE 2.9 : DIAGRAMME DE COMMUNICATION17

FIGURE 2.10 : LA VUE ('4+1')18

FIGURE 2.11 : CAS D'UTILISATION UML PILOTE UP.....19

FIGURE 2.12 : PRÉSENTATION DE CYCLE DE VUE PROCESSUS.....20

CHAPITRE 3

FIGURE 3.1 : CAS D'UTILISATION POUR L'ADMINE.....26

FIGURE 3.2 : CAS D'UTILISATION POUR ETUDIANT.....27

FIGURE 3.3 : CAS D'UTILISATION POUR ENSEIGNANT.....27

<u>FIGURE 3.4 : DIAGRAMME DE CLASSE DU DOMAINE.....</u>	<u>28</u>
<u>FIGURE 3.5 : DIAGRAMME DE SEQUENCE CONNEXION D'ÉTUDIANTS.....</u>	<u>29</u>
<u>FIGURE 3.6 : DIAGRAMME DE SEQUENCE AJOUTER LES EMPLOIS DES TEMPS.....</u>	<u>30</u>
<u>FIGURE 3.7: DIAGRAMME DE SEQUENCE AJOUTER LES ETUDIANTS.....</u>	<u>31</u>
<u>FIGURE 3.8: DIAGRAMME DE SEQUENCE D'AJOUTER DEPARTEMENTS.</u>	<u>32</u>
<u>FIGURE 3.9 : DIAGRAMME DE SEQUENCE CONNEXION D'ADMINE.....</u>	<u>33</u>
<u>FIGURE 3.10 : DIAGRAMME DE SEQUENCE AJOUTER LES COURS.</u>	<u>34</u>
<u>CHAPITRE 4 :</u>	
<u>FIGURE 4.1 : INTERFACE D'APPSERV.....</u>	<u>38</u>
<u>FIGURE 4.2 : PRÉSENTATION DE LA BASE DE DONNÉES.....</u>	<u>41</u>
<u>FIGURE 4.3 : DOSSIER DE SITE.....</u>	<u>42</u>
<u>FIGURE 4.4 : ARBORESCENCE DU SITE.....</u>	<u>42</u>
<u>FIGURE 4.5 : CAPTURE PAGE D'ACCUEIL.....</u>	<u>43</u>
<u>FIGURE 4.6 : CAPTURE DE PAGE ADMINE OU ENSINGANT</u>	<u>44</u>
<u>FIGURE 4.7 : FIGURE 4.3CAPTURE PAGE DE PROFILE ADMINE.....</u>	<u>44</u>
<u>FIGURE 4.8 : CAPTURE FORMULAIRE AJOUTER DES ENSINGANTS</u>	<u>45</u>

Bibliographie

- [1] **Livre:** .NET par Dick Lantin-édition © Groupe Eyrolles, 2003
- [2] **Livre:** Premiers pas en CSS et HTML Guide pour les débutants. F. DRAILLARD, Eyrolles, 2006.
- [3] **Livre:** JavaScript by Example, Ellie Quigley, Prentice Hall, 2003.
- [4] **Livre:** Professional ADO.NET 2, Wiley Publishing, Inc, 2006.

webographie

- [1] **Site:** <http://www-igm.univ-mlv.fr/~dr/XPOSE2002/Tedeschi/>
- [2] **Site:** <http://asp.developpez.com/cours/>
- [3] **Site:** Wikipédia(l'encyclopédie en ligne): <http://www.wikipedia.org/>
- [4] **Site:** MSDN: <http://msdn.microsoft.com/fr-FR/>