

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



*Mémoire de fin de cycle
en vue d'obtention du diplôme de master professionnel en informatique
option : Administration et sécurité des réseaux*

Thème

Conception et réalisation d'une application web pour la
gestion des étudiants d'une école privée.
Cas d'étude : "ISA School"

Mémoire soutenu par :

M^{elle} SAICHE Cylia
M^r OUYOUGOUTE Abdelatif

Devant le jury composé de :

Président :	<i>M^r LARBI</i> Ali	M.A.A, U.A.M Béjaïa
Examineur :	<i>M^r AKILAL</i> Abdellah	M.A.B, U.A.M Béjaïa
Examineur :	<i>M^{elle} ALBANE</i> Saadia	Doctorante, U.A.M Béjaïa
Encadreur :	<i>M^{me} ZIDANI</i> Ferroudja	M.A.A, U.A.M Béjaïa
Co-Encadreur :	<i>M^{elle} BRAHIMI</i> Samira	Ingénieur, U.A.M Béjaïa

Promotion 2014/2015

Remerciements

En premier, nous aimerions remercier le bon Dieu le tout puissant de nous avoir donné le courage et la volonté de réaliser ce projet.

Nous désirons remercier nos chers parents qui nous ont soutenus et encouragé durant toute notre vie et pendant notre cursus d'étude.

Nos remerciements les plus chaleureux vont à Mme F.ZIDANI et Mlle S.BRAHIMI pour leurs disponibilités et leurs très précieux conseils ainsi que leurs remarques qui nous ont permis d'améliorer la qualité de ce travail.

Nous tenons à exprimer toute notre grande gratitude aux membres de jury d'avoir accepté de juger ce travail.

Nous remercions Mlle BENKERROU Hayet, et toutes les personnes ayant contribué de près ou de loin à l'élaboration de ce modeste travail.

Nos vifs remerciements s'adressent également à tous nos enseignants de la faculté sciences exactes de l'université A/MIRA Bejaia pour la formation qu'ils ont eu le soin de nous apporter le long de notre cursus universitaire.

Dédicaces

A mes adorables parents que j'aime énormément

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

A ma très chère petite sœurs Asma et mes chers frères Fateh et Lounis

Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès.

A ma chère grande sœur Mounia son mari et sa belle famille et son merveilleux et adorable garçon Fayez.

A mes grands parents Larbi et Zahra, que dieu les garde.

A la mémoire de mes grands parents Lakhdar et Baya et Dahbiya, et mon oncle Lyazid, que dieu les accueille dans son vaste paradis.

A tous mes oncles et tantes et leurs familles et enfants.

A mon camarade **Lotfi** pour son soutien et sa disponibilité son aide et ses conseils durant la réalisation de ce travail, sans oublié sa famille.

A tous mes aimables amis et camarades.

A toute la promotion d'informatique 2015 spécifiquement le groupe 4 et tous ceux qui me connaissent .

SAICHE Cylia

Dédicaces

Je dédie ce travail premièrement a mes chères parents

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi.

Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

A ma chère soeur Chayma et mes frères Abderahime et Zakaria

Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès.

A la mémoire de mes grands parents, que dieu les accueille dans son vaste paradis.

A tous mes oncles et tantes et leurs familles et enfants.

Tous mes aimables amis et camarades.

A ma camarade **Cylia** pour sa disponibilité et compréhension et aide durant la réalisation de ce travail et sa famille.

Tous ceux qui m'ont aidé de près ou de loin à réaliser ce projet.

Tous ceux qui me connaissent.

OUYOUGOUTE Abdelatif

Table des matières

Introduction générale	1
1 Les technologies web	3
1.1 Introduction	3
1.2 Les technologies du web	3
1.2.1 Un site web	3
1.2.2 Une application web	3
1.2.3 L'architecture des applications web	4
1.2.4 L'application Client/serveur	5
1.2.5 Le serveur web	5
1.2.6 Le fonctionnement d'une application web	5
1.2.7 L'hébergement d'une application web	6
1.2.8 Les langages utilisés pour la création des applications web	7
1.3 La présentation de l'organisme d'accueil	7
1.3.1 Plan et effectif d'ISA School	7
1.4 Conclusion	8
2 Etude préalable	9
2.1 Introduction	9
2.2 La description du contexte du projet	9
2.2.1 Le dossier étudiant	9
2.2.2 Les objectifs de la gestion des étudiants	10
2.2.3 La problématique	10
2.3 Les méthodes d'analyse et de conception	11
2.3.1 Le 2 Track Unified Process	11
2.3.2 Le processus unifié	11
2.3.3 Le choix de la méthode	13
2.4 L'identification des besoins	13
2.4.1 Les besoins fonctionnels	13
2.4.2 Les besoins non fonctionnels	13
2.5 L'analyse des besoins	13
2.5.1 L'identification des acteurs	14
2.5.2 L'identification des cas d'utilisations de l'application	14
2.5.3 Le diagramme des cas d'utilisations globale	15
2.6 Le diagramme des séquences	24
2.7 Conclusion	30

3	Conception	31
3.1	Introduction	31
3.2	Diagramme d'interaction	31
3.2.1	Diagramme d'interaction authentification d'un utilisateur	32
3.2.2	Diagramme d'interaction gérer un dossier étudiant	33
3.3	Diagramme des classes	36
3.3.1	Description des classes	36
3.4	Passage au relationnel	40
3.4.1	Règles de passage du diagramme de classes au modèle relationnel	40
3.5	Conclusion	41
4	Réalisation	42
4.1	Introduction	42
4.2	Environnement et outils de développement de notre application	42
4.3	Structure et maquette de l'application	44
4.3.1	Maquette de la page d'accueil après l'authentification	45
4.3.2	Maquette de la page d'ajout d'un étudiant	45
4.4	Présentation de quelques interfaces de l'application	46
4.4.1	Capture d'écran : s'authentifier	46
4.4.2	Capture d'écran : Page d'accueil	47
4.4.3	Capture d'écran :ajouter un dossier étudiant	48
4.4.4	Capture d'écran : ajouter un dossier enseignant avec champs vide	49
4.4.5	Capture d'écran : ajouter un groupe existant	50
4.4.6	Capture d'écran : modifier un dossier étudiant	51
4.4.7	Capture d'écran : supprimer un dossier étudiant	52
4.4.8	Capture d'écran : confirmer la suppression	53
4.5	Conclusion	53
	Conclusion générale et perspectives	54
	Annexes a	lv
	Annexes b	lvi
	Annexes c	lvii
	Bibliographie	lviii

Table des figures

1.1	Architecture 3-tiers d'une application web	4
1.2	Fonctionnement d'une application web [15].	6
2.1	Diagramme des cas d'utilisation globale	15
2.2	Diagramme de cas d'utilisation s'authentifier	16
2.3	Diagramme de cas d'utilisation gérer les utilisateurs	17
2.4	Diagramme de cas d'utilisation gérer les étudiants	18
2.5	Diagramme de cas d'utilisation gérer les enseignants	19
2.6	Diagramme de cas d'utilisation gérer les groupes	20
2.7	Diagramme du cas d'utilisation consulter un dossier étudiant	21
2.8	Diagramme de cas d'utilisation gérer paiement	22
2.9	Diagramme de cas d'utilisation gérer les versements des étudiants	22
2.10	Diagramme de cas d'utilisation imprimer un document	23
2.11	Diagramme de cas d'utilisation calculer les statistiques	23
2.12	Diagramme séquence système lié au cas d'utilisation "s'authentifier"	25
2.13	Diagramme séquence système lié au cas d'utilisation "ajouter un dossier étudiant"	26
2.14	Diagramme séquence système lié au cas d'utilisation "modifier un dossier étudiant"	27
2.15	Diagramme séquence système lié au cas d'utilisation "supprimer un dossier étudiant"	28
2.16	Diagramme séquence système lié au cas d'utilisation "consulter un étudiant"	29
2.17	Diagramme séquence système lié au cas d'utilisation "imprimer un document"	30
3.1	Diagramme d'interaction authentification d'un utilisateur	32
3.2	Diagramme d'interaction ajouter un dossier étudiant	33
3.3	Diagramme d'interaction modifier un dossier étudiant	34
3.4	Diagramme d'interaction supprimer un dossier étudiant	35
3.5	Diagramme des classes	37
4.1	Etapas d'une demande de page HTML doté d'un code JavaScript	43
4.2	Structure de l'application	44
4.3	maquette de la page d'accueil	45
4.4	maquette d'ajout d'un étudiant	45
4.5	Page authentification	46
4.6	Page d'accueil	47
4.7	Page ajouter un dossier étudiant	48
4.8	Page ajouter un dossier enseignant	49
4.9	Page ajouter un groupe	50
4.10	Page modifier un dossier étudiant	51
4.11	Page supprimer un dossier étudiant	52

4.12 Page confirmer la suppression	53
--	----

Liste des tableaux

- 3.1 Description des classes de l'application 36
- 3.2 Dictionnaire des données 39

Liste des abréviations

UP	unified processor ou processus unifié.
2TUP	Two Track unified processor
PHP	Hypertexte PréProcesseur ou Personal Home Page.
CSS	Cascading Style Sheets.
XML	eXtensible Markup Langage.
HTML	Hypertext Markup Langage.
XHTML	eXtensible Hypertext Markup Langage.
WWW	Word Wide Web.
UML	Unified Modeling Langage.
BDD	Base De Données.
SGBD	Serveur de Gestion de Base de Données.
SQL	Structured Query Language
MySQL	My Structured Query Language
JPEG	Joint Photographie Expert Group
PNG	Protocol Network Graphics
HTTP	Hyper Text Transfert Protocol

Introduction générale

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela à l'aide de l'informatique, qui joue un rôle important dans le développement de nombreuses entreprises et organisations.

Avant l'invention de l'ordinateur, nous enregistrons toutes les informations manuellement sur des supports en papier. Ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières [1].

La nouvelle logique de l'organisation du travail demande aux établissements d'éducatifs et d'apprentissages d'utiliser essentiellement l'information comme matière première pour pouvoir être plus efficace. Ils doivent donc intégrer un développement du système d'information dans leurs investissements stratégiques, dans la mesure où ils structurent la saisie, le stockage, l'organisation et la communication de l'information.

Aujourd'hui, les écoles privées auxquelles nous rattacherons d'ailleurs notre étude, font partie intégrante des établissements scolaires où l'informatique pourra aider. En effet, la croissance de la population étudiante nécessite la mise en place d'une gestion rationnelle et rapide, or et jusqu'à ce jour, la manière de gérer manuellement est encore dominante d'où la nécessité d'introduire l'informatique dans ces centres de formations.

Problématique

Nous avons pu constater, en effet, pendant notre observation au sein de l'école des langues étrangères "ISA School" qu'une majeure partie des traitements se fait manuellement, ce qui engendre un certain nombre de problèmes tels que :

1. Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l'établissement des documents.
2. Recherche difficile sur les registres qui engendre une perte de temps.
3. Insécurité des informations.
4. Possibilité d'erreur dans le remplissage des différents documents et registres.
5. Possibilité d'erreur dans le calcul des statistiques.

Donc, la meilleure solution pour pallier aux problèmes est l'informatisation afin, d'assurer l'accès instantané aux données et une sécurisation de ces dernières, ce qui simplifie le travail administratif. De ce fait, nous avons proposé aux responsables de l'école de leur concevoir une application pour la gestion de leur école.

L'objectif de notre projet présenté dans ce rapport est la conception et la réalisation d'une application web simple de gestion des entrées/sorties des étudiants ainsi que les enseignants dans un centre de formation. Et pour la réalisation de cette tâche, notre choix s'est porté sur la méthode de développement logiciel UP (Processus Unifié) et UML (Unified Modeling Language) comme langage de modélisation.

Ce présent travail sera structuré en quatre chapitres :

Dans le premier chapitre, nous présenterons comme première partie quelques notions sur les technologies web, et comme deuxième partie la présentation de l'organisme d'accueil.

Dans le second, nous présenterons la problématique et nous donnerons l'objectif de notre travail et la spécification des besoins.

Dans le troisième chapitre, nous allons faire une conception détaillée de notre application en présentant les cas d'utilisation sous différents diagrammes.

Le quatrième et dernier chapitre, sera la partie de l'implémentation et la réalisation de notre application, où nous présenterons l'environnement de développement, la structure de notre application et quelques interfaces de celle si.

Enfin, nous clôturons ce mémoire par une conclusion générale et perspective.

Les technologies web

1.1 Introduction

Le développement des applications web présente certaines particularités, au niveau technique et ergonomique. Cette spécificité nous oblige, au moment de la conception, à préconiser des méthodes de conception et des méthodes de travail dédiées à ce genre d'applications [15].

Ce chapitre est composé de deux sections, la première est consacrée pour définir quelques généralités sur les technologies web ; la deuxième a mis l'accent sur la présentation de l'organisme d'accueil.

1.2 Les technologies du web

Dans ce qui suit, nous définirons un site web, une application web, et nous présenterons d'une manière brève la différence entre les deux.

1.2.1 Un site web

Un site web est composé d'un ensemble de documents structurés, nommés pages web, stockés (hébergés) sur un ordinateur (serveur) connecté au réseau mondial [3]. Il existe deux types de sites web :

Un site statique : constitué des pages web avec du contenu fixe. Chaque page est codée en HTML (HyperText Markup Language) et affiche les mêmes informations à chaque visiteur. La modification de ces pages se fait par une intervention de l'administrateur du site web [26].

Un site dynamique : les pages du site qui le compose peuvent être modifiables par les visiteurs. De plus, entre deux visites sur un même site, le contenu de la page peut être différent sans action de l'administrateur du site Internet. Le serveur qui fait fonctionner le site utilise une technologie de Scripting (comme PHP : Hypertext Preprocessor) ainsi qu'une base de données comme MySQL(My Structured Query Language)[26].

1.2.2 Une application web

Une application web est un ensemble de pages qui interagissent avec les utilisateurs, les unes avec les autres, ainsi qu'avec les différentes ressources d'un serveur web, notamment les bases de données [4].

L'architecture d'une application web contient un serveur d'application qui rend possible le traitement d'une logique et d'un état applicatif. tandis que l'architecture d'un site web contient trois composants principaux [15] :

- Un serveur web.
- Une connexion réseau.
- Des navigateurs clients.

En d'autres termes une application web sert à manipuler l'information et elle est soumise à des contrôles d'accès et de rôles, et un site web sert à exposer l'information [23].

1.2.3 L'architecture des applications web

Une architecture est un modèle générique et conceptuel qui se rapporte à un sujet et qui représente la fonctionnalité, la structure, le positionnement, l'interrelation des différents types d'éléments (hardware, logiciels, infrastructure) qui la composent [2].

En règle générale, une application est découpée en 3 niveaux (couches) d'abstraction [17] :

La couche présentation : c'est la partie de l'application visible par les utilisateurs (nous parlerons d'interface utilisateur). Dans notre cas, cette couche est un navigateur web, qui se présente sous forme de pages HTML, composée de formulaire et de bouton.

La couche métier : correspond à la partie fonctionnelle de l'application, celle qui implémente la logique, et qui décrit les opérations que l'application opère sur les données, en fonction des requêtes d'un utilisateur effectuées au travers de la couche présentation.

La couche accès aux données : elle consiste en la partie gérant l'accès à la base de données du système.

Il existe différentes architectures pour une application web :

- Architecture 1-tiers
- Architecture 2-tiers
- Architecture 3-tiers
- Architecture n-tiers[17].

Dans notre application nous avons utilisé une architecture à 3-tiers qui est illustrée dans la figure (1.1) sise ci-dessous :

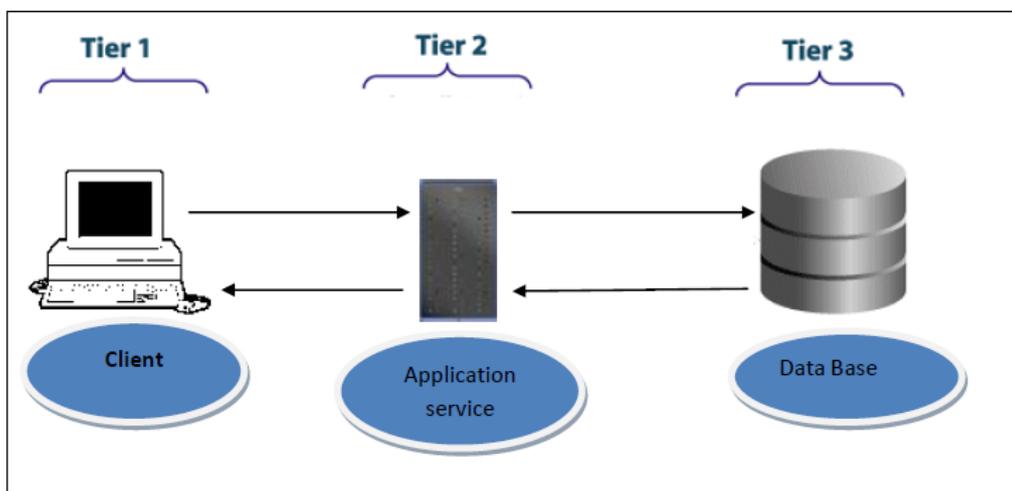


FIGURE 1.1 – Architecture 3-tiers d'une application web

1.2.4 L'application Client/serveur

Une application web est dite basée sur un modèle Client - Serveur si la base de données se trouve sur un serveur qui ne sert qu'à ça, et pour interagir avec cette base de données, il faut utiliser un logiciel "client" qui va interroger le serveur et transmettre la réponse que le serveur lui aura donnée [27].

Le serveur peut être installé sur une machine différente du client.

Par conséquent, lorsque nous installons un SGBD(système de gestion de base des données), nous installons en réalité deux choses :

- le serveur
- le client

Chaque requête (insertion/modification/lecture de données) est faite par l'intermédiaire du client. Nous pouvons donc avoir besoin d'un langage pour discuter avec le client, pour lui donner les requêtes que nous souhaitons effectuer.

Dans le cas de notre application nous avons MySQL comme système de gestion des bases de données, et le SQL(Structured Query Language) comme langage de système de gestion des bases de données.

1.2.5 Le serveur web

Un serveur web est un logiciel qui fournit des pages web en réponse à des requêtes de navigateurs web. Une requête de page est générée lorsqu'un utilisateur clique sur un lien d'une page web ou saisit une URL (localisateur uniforme de ressources) dans le champ adresse du navigateur [10].

1.2.6 Le fonctionnement d'une application web

Lorsqu'un serveur web reçoit une requête de page web statique, il transmet simplement cette page au navigateur requérant. En revanche, lorsque le serveur web reçoit une requête de page dynamique, il transmet cette page à une extension logicielle spéciale chargée d'achever la page. Ce logiciel spécial est appelé serveur d'application.

Le serveur d'application lit le code de la page, termine cette page en fonction des instructions figurantes dans le code, puis en retire le code.

Il en résulte une page statique que le serveur d'application renvoie au serveur web, lequel transmet alors cette page au navigateur requérant.

Le navigateur reçoit uniquement du code HTML pur lorsque la page lui est transmise.

Un serveur d'application nous permet de travailler avec des ressources côté serveur telles que les bases de données. Une page dynamique peut, par exemple, ordonner au serveur d'application d'extraire des données de la base de données et de les insérer dans le code HTML de la page.

L'instruction d'extraction des données de la base est nommée requête de base de données. Une requête est composée de critères de recherches rédigés dans un langage de base de données appelé SQL. La requête SQL est rédigée dans les scripts ou les balises côté serveur de la page [15].

Toutes ces actions sont résumées dans le schéma présenté dans la figure (1.2) sise ci-dessous :

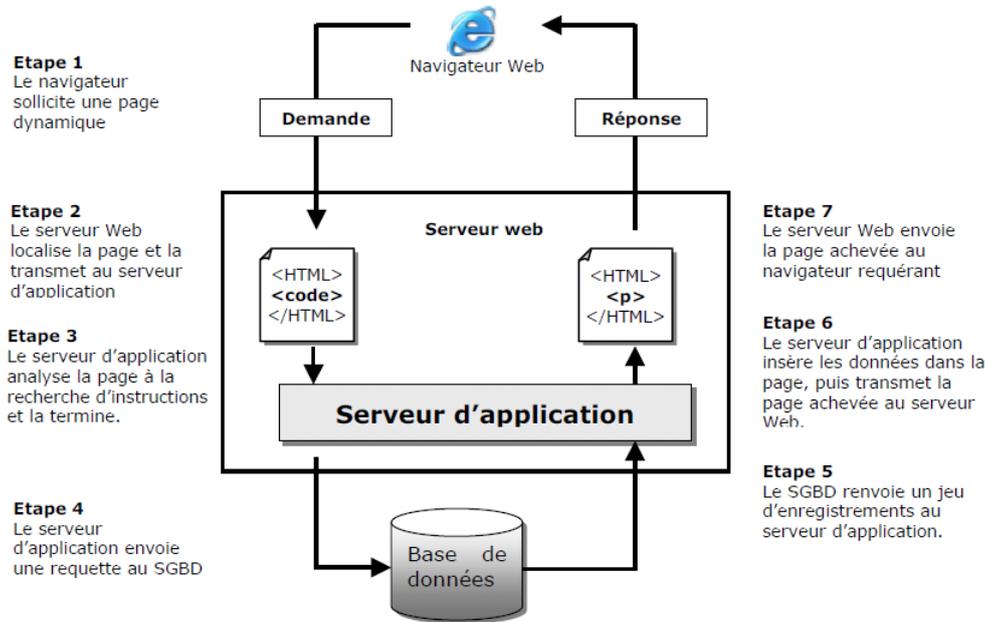


FIGURE 1.2 – Fonctionnement d'une application web [15].

Généralement et après la réalisation et l'implémentation d'une application ; une dernière étapes reste à accomplir c'est l'hébergement de l'application afin qu'elle devienne accessible et manipulable par les utilisateurs.

1.2.7 L'hébergement d'une application web

Définition L'hébergement est un service visant à rendre un site ou une application web accessible sur Internet.

Afin que les pages soient visibles par tout le monde, il faut qu'elles soient stockées sur un ordinateur connecté en permanence à l'internet (serveur)[14][9].

Un hébergeur est comme une société spécialisée dans l'hébergement de sites web sur des serveurs, sa principale activité sur internet consiste à installer des serveurs, à les sécuriser, à les tenir à jour en installant les mises à jour de sécurité pour éviter les attaques malveillantes et à les réparer en cas de panne [8].

Il existe plusieurs types d'hébergements dont :

1. Hébergement gratuit : ce type d'hébergement offre un service basique, avec un espace de stockage assez limité, avec une fiabilité qui n'est pas toujours optimale.
2. Hébergement payant : ce type d'hébergement propose un service de qualité, avec de bonnes capacités de stockage ainsi qu'une grande fiabilité. Le support sera également plus disponible et plus réactif en cas de problème et aidera avec des réponses personnalisées.

Dans la partie suivante nous parlerons des langages utilisés pour la création des applications.

1.2.8 Les langages utilisés pour la création des applications web

Le développement d'une applications web nécessite la connaissance des différents langages utilisés dans les technologies web : HTML pour la présentation des pages, CSS (Cascading Style Sheets) pour la charte graphique, JavaScript, Java ou ActionScript pour les automatismes exécutés par le client, ainsi qu'un langage tel que Java, le PHP et d'autre [7].

Le langage PHP

Le PHP est un langage de scripts multi plateformes, orienté-objet très répandu. Il a été conçu pour permettre la création des applications dynamiques, la génération du code (HTML, XHTML (Extensible HyperText Markup Language), CSS) et des données (JPEG(Joint Photographic Experts Group),PNG(Portable Network Graphics)).Il est le plus souvent couplé à un serveur Apache, bien qu'il peut être installer sur d'autre serveurs Http(HyperText Transfer Protocol)[5].

Les instructions du PHP sont lues et analysées par le serveur web. Elles ne parviennent jamais jusqu'au navigateur qui affiche la page car le serveur web remplace le code PHP par le contenu que le code avait pour but de générer.

Etant donnés tous ces éléments et les contraintes relatives à ce projet, nous avons choisi d'utiliser le langage PHP, qui présente un bon compromis entre la rapidité de mise en place et les performances.

D'autre part, une application web avec PHP est très facilement déplorable et fonctionne avec tous les systèmes d'exploitation, que ce soit Mac, Linux ou Windows [16].

Après avoir vue les différentes technologies web dans la première partie, nous allons passer à la deuxième partie qui est l'organisme d'accueil.

1.3 La présentation de l'organisme d'accueil

L'école des langues étrangères ISA School est un centre de formation pour l'apprentissage des langues (français, anglais, allemand, espagnol, italien...) créé en 2007.

Ce centre de formation vise à apporter le meilleur suivi linguistique possible aux étudiants ainsi que le personnel de tout niveaux.

ISA School reçoit chaque année un nombre considérable d'étudiants qui sont orienté au sein de cet établissement vers un choix de langue spécifique, et à des heures d'apprentissages presque convenable à tout le monde.

L'école se constitue de 4 sièges répartie sur des endroits différents au niveau de la wilaya de Bejaia, la direction se trouve à Elkseur.

1.3.1 Plan et effectif d'ISA School

L'école ISA School, est équipé de :

1. Une salle d'attente.
2. des salles pour les cours.

3. Un bureau pour le directeur.
4. Un bureau pour chaque gérants.
5. Un bureau pour la réception.
6. Une salle de réunion.
7. Une salle contenant tous les archives depuis sa création a nos jours.
8. deux salles équipées de micro-ordinateur et d'outils d'audio et visualisation.
9. Une salle de sanitaire.

Le centre de formation comporte :

- plusieurs enseignants de différentes langues de tous niveaux confondus,
- un matériel informatique assez important réparti d'une manière équitable entre les différents bureaux.

1.4 Conclusion

Dans la première section de ce chapitre, nous avons vu quelques notions sur les technologies web tel que les applications web, et nous avons présenté le langage PHP comme notre choix de langage de programmation.

Dans la deuxième section nous avons présenté l'organisme d'accueil de l'école des langues étrangères ISA School.

Dans le chapitre suivant nous décrirons le contexte de notre projet et nous définirons les cas d'utilisations de notre application.

Etude préalable

2.1 Introduction

Dans ce chapitre, nous décrirons le contexte du projet, nous poserons la problématique et nous exposerons les objectifs de notre travail ; puis nous passerons aux méthodes de conception et à l'identification des besoins et nous terminerons par la présentation des diagrammes des cas d'utilisations et de séquences de notre application.

2.2 La description du contexte du projet

La gestion des étudiants c'est l'admission de ces derniers pour des formations en langues étrangères, ainsi que la gestion de leurs mouvements au sein de l'école.

Lors de leurs entré, les étudiants sont accueilli au sein de l'école, en fonction de leurs choix de langues.

L'étudiant est un élément essentiel pour l'existence et le fonctionnement du centre de formation, où il doit avoir un dossier propre à lui.

Le but essentiel de la gestion des étudiants est le suivi et l'historique, et l'intérêt de l'informatisation est :

- d'avoir un dossier numérique pour chaque étudiant.
- de planifier des inscriptions et les durées des formations (date début et date de fin de formation).
- de contrôler les paiements des frais des formations.

2.2.1 Le dossier étudiant

Un dossier étudiant est la mémoire dont laquelle sont consignées toutes les informations nécessaires à la prise en charge et à la surveillance d'un étudiant. En d'autre terme, un dossier étudiant englobe tout ce qui peut être mémorisé chez un étudiant.

Compte tenu de ce rôle, le dossier étudiant reste toujours l'outil principal de centralisation et de coordination de l'activité éducatif.

Après la description du contexte de notre projet, dans ce qui suit nous passerons aux objectifs de la gestion.

Juste après nous poserons la problématique et nous exposerons par la suite les objectifs de notre projet.

2.2.2 Les objectifs de la gestion des étudiants

La gestion des étudiants a pour objectif de

1. Savoir traiter les demandes d'accès aux dossiers des étudiants.
2. Améliorer la gestion des informations en les centralisant.
3. Constituer une banque de données.
4. Accéder à l'historique des étudiants précédents, formation prise...
5. Identifier les règles de gestion du dossier étudiant.
6. Etablir les documents qualité relatifs à la gestion du dossier.
7. Améliorer la disponibilité du dossier et des informations qu'il contient, y compris en cas d'urgence.

2.2.3 La problématique

L'école désire synchroniser ses services d'une manière performante, facile et efficace afin de satisfaire le client et garantir le maintien de l'activité pour détecter les problèmes existants. Nous avons interrogé le directeur de l'école qui nous a cité quelques anomalies, mais pour localiser leur source, nous nous sommes mis en pratique avec elles. Et après une observation continue, nous avons pu observer les insuffisances suivantes :

1. Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l'établissement des documents.
2. Recherche difficile sur les registres qui engendre une perte de temps.
3. Insécurité des informations.
4. Possibilité d'erreur dans le remplissage des différents documents et registres.
5. Possibilité d'erreur dans le calcul des statistiques.
6. Nombre important des archives qui engendre une difficulté de stockage.
7. Détérioration des archives à force de leur utilisation trop fréquente.
8. Mauvaise codification sur quelques objets dans la gestion d'information.
9. Difficulté dans la gestion des différents sièges réparti dans des lieux différents.

Nous constatons que la solution informatique est la plus adéquate puisqu'elle répond mieux aux anomalies souvent fréquentées dans la gestion manuelle.

Ainsi nous avons décidé de concevoir une application web qui va gérer les activités de cette école et qui va permettre par la suite de minimiser le support papier et d'améliorer la rapidité de l'accès à l'information. Et pour cela nous avons assigné à notre étude les objectifs suivants :

1. Rapidité dans l'établissement des différents documents.
2. Facilité de la recherche et l'accès aux informations.
3. Stockage des informations sur des supports informatiques ce qui assurera leur sécurité.
4. Gain de temps dans le calcul des statistiques.
5. Automatiser les tâches qui se traitent manuellement.
6. Proposer une bonne codification.

Nous arrivons maintenant à la deuxième partie de ce chapitre, où nous présenterons l'analyse de notre application.

2.3 Les méthodes d'analyse et de conception

Une méthode d'analyse et de conception a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système, afin de rendre ce développement plus fidèle aux besoins du client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc.

La phase de conception permet de décrire de manière non ambiguë, en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Il existe plusieurs méthodes d'analyses, tel que 2TUP (Two Track Unified Process) et UP (Unified Process).

2.3.1 Le 2 Track Unified Process

2TUP est un processus de développement logiciel qui implémente le processus unifié. 2TUP est piloté par les risques, et utilise UML comme langage de modélisation car il y a une correspondance entre ses étapes et les différents diagrammes d'UML ; son idée de base est : toute évolution imposée pour le système peut se décomposer et se traiter parallèlement, suivant deux axes (" 2 tracks ") [22] :

- Un axe fonctionnel
- Un axe technique

La réalisation du système 2TUP consiste à fusionner les résultats des deux branches en forme de Y :

1. La branche fonctionnelle : à ce niveau se fait la capture des besoins fonctionnels et l'analyse.
2. La branche technique : à ce niveau se fait la capture des besoins techniques et la conception générique.
Cette étape permet de minimiser l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles.
3. La branche du milieu : à ce niveau se fait la conception préliminaire, la conception détaillée, l'étape de codage et l'étape de recette [20].

Note : pour plus de précision veuillez consulter l'annexe (c).

2.3.2 Le processus unifié

Le processus unifié est une méthode générique de développement logiciel, son principal intérêt est de limiter les risques lors du développement des projets [29][6].

a. Les activités de développement du processus unifié [29]

Les activités de développement de l'UP sont dérivées par cinq disciplines fondamentales qui sont [29] :

1. la capture de besoins,
2. l'analyse et la conception,
3. l'implémentation,
4. le test,

5. le déploiement.

b. Les caractéristiques du processus unifié [22]

1. **Centré sur l'architecture** : l'architecture peut être considérée comme l'ensemble de vues du système qui vont provenir des besoins de l'entreprise et des différents intervenants.
2. **Piloté par les cas d'utilisation** : le modèle des cas d'utilisations guide le processus unifié et décrit les fonctionnalités du système.
3. **Itératif et incrémental** : les itérations se succèdent dans un ordre logique permettant de donner lieu à un incrément et donc d'établir un développement plus optimisé et efficace.

c. Les phases du cycle de vie du processus unifié

1. Incubation/ création
2. Elaboration
3. Construction
4. Transition

d. Le langage de modélisation du processus unifié

L'UP s'appuie sur UML qui est un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vues [19].

UML définit 13 types de diagrammes devisés en deux catégories[24] :

Les diagrammes structurels Les diagrammes structurels ou statiques rassemblent :

1. Diagramme de classes (Class diagram)
2. Diagramme d'objets (Object diagram)
3. Diagramme de composants (Component diagram)
4. Diagramme de déploiement (Deployment diagram)
5. Diagramme des paquetages (Package diagram)
6. Diagramme de structure composite (Composite Structure Diagram)

Les diagrammes comportementaux Les diagrammes comportementaux rassemblent :

1. Diagramme des cas d'utilisation
2. Diagramme états-transitions (State Machine Diagram)
3. Diagramme d'activité (Activity Diagram)
4. Diagramme de séquence (Sequence Diagram)
5. Diagramme de communication (Communication Diagram)
6. Diagramme global d'interaction (Interaction Overview Diagram)
7. Diagramme de temps (Timing Diagram)

Note :pour plus de précision veuillez consulter l'annexe (b).

2.3.3 Le choix de la méthode

Pour la réalisation de notre application, notre choix a été porté sur le Processus Unifié. En effet, le processus unifié est une solution de développement logiciel adaptée à tout type de projet.

Ces traits distinctifs tiennent compte de trois notions : piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental.

Le langage de modélisation que nous avons utilisé est UML, qui est une partie intégrante de la démarche UP. Ces diagrammes sont largement utilisés dans chaque étape et phase de ce processus de développement.

2.4 L'identification des besoins

UP distingue deux types de besoins :

1. les besoins fonctionnels qui conduisent à l'élaboration des cas d'utilisation ;
2. Les besoins techniques (non-fonctionnels) qui aboutissent à la rédaction d'une des exigences de système pour sa réalisation et son bon fonctionnement.

2.4.1 Les besoins fonctionnels

1. Avoir une base de données pour le stockage des étudiants et des enseignants.
2. Manipulation et mise à jour de la base de données.
3. Le calcul des statistiques selon les versements des étudiants et les paiements des enseignants.
4. L'application conçue devra fonctionner en mode 3 - tiers (client, serveur de données, serveur d'application).
5. Chaque utilisateur possède un login et un mot de passe unique pour accéder à cette application.
6. Le gérant peut ajouter, supprimer, modifier ses données, gérer ses étudiants, enseignants et les groupes tandis que l'agent d'accueil n'as que le droit de les consulter.
7. Avoir un historique des paiements des enseignants et des versements des étudiants.

2.4.2 Les besoins non fonctionnels

- L'application doit permettre de gérer les accès des utilisateurs selon un privilège et un état d'activation de chaque compte.
- Il faut garantir la sécurité d'accès à l'espace administrateur afin de protéger les données personnelles des utilisateurs.
- Prévenir contre l'accès direct avec les liens URL et définir un délai de temps pour la fermeture de session non active.
- L'interface de cette application doit être ergonomique, conviviale et voire même apte à aider l'utilisateur à mieux gérer son espace de travail.

2.5 L'analyse des besoins

Nous traitons dans cette partie l'analyse fonctionnelle de notre projet. D'abord nous identifions les acteurs impliqués, ensuite nous spécifions les cas d'utilisations de notre application.

2.5.1 L'identification des acteurs

Les différents acteurs qu'on va trouver dans nos diagrammes de cas d'utilisation sont :

1. **gérant/directeur** : l'acteur principal dans notre application, c'est lui qui
 - reçoit l'étudiant et les enseignants,
 - s'occupe de la rédaction de leurs dossiers, ainsi que le suivi dans l'école,
 - reçoit des versements et effectue des payes,
 - oriente l'étudiant vers un choix de langue s'il est nécessaire
 - gère et classe les groupes.
2. **Agent d'accueil** : comme son nom le désigne il accueille l'étudiant et il le droit de consulter.
3. **Administrateur de site** : est le responsable de la gestion depuis la conception jusqu'à la maintenance de l'application web.

Note : en terme de besoins nous avons les remarques suivantes :

- Le directeur est un acteur qui peut faire toutes les tâches du gérant, mais durant tout le travail nous utiliserons que l'acteur 'gérant'.
- L'acteur 'utilisateur' c'est l'ensemble des acteurs (gérant/directeur, administrateur, agent d'accueil).
- Le gérant peut faire les tâches de l'administrateur de site.

2.5.2 L'identification des cas d'utilisations de l'application

Définition

Le diagramme des cas d'utilisations identifie les fonctionnalités fournies par le système, les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers [21].

Un cas d'utilisation correspond à un certains nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur [25].

La figure 2.1 sise ci-dessous présente le digramme globale des cas d'utilisation de notre application.

2.5.3 Le diagramme des cas d'utilisations globale

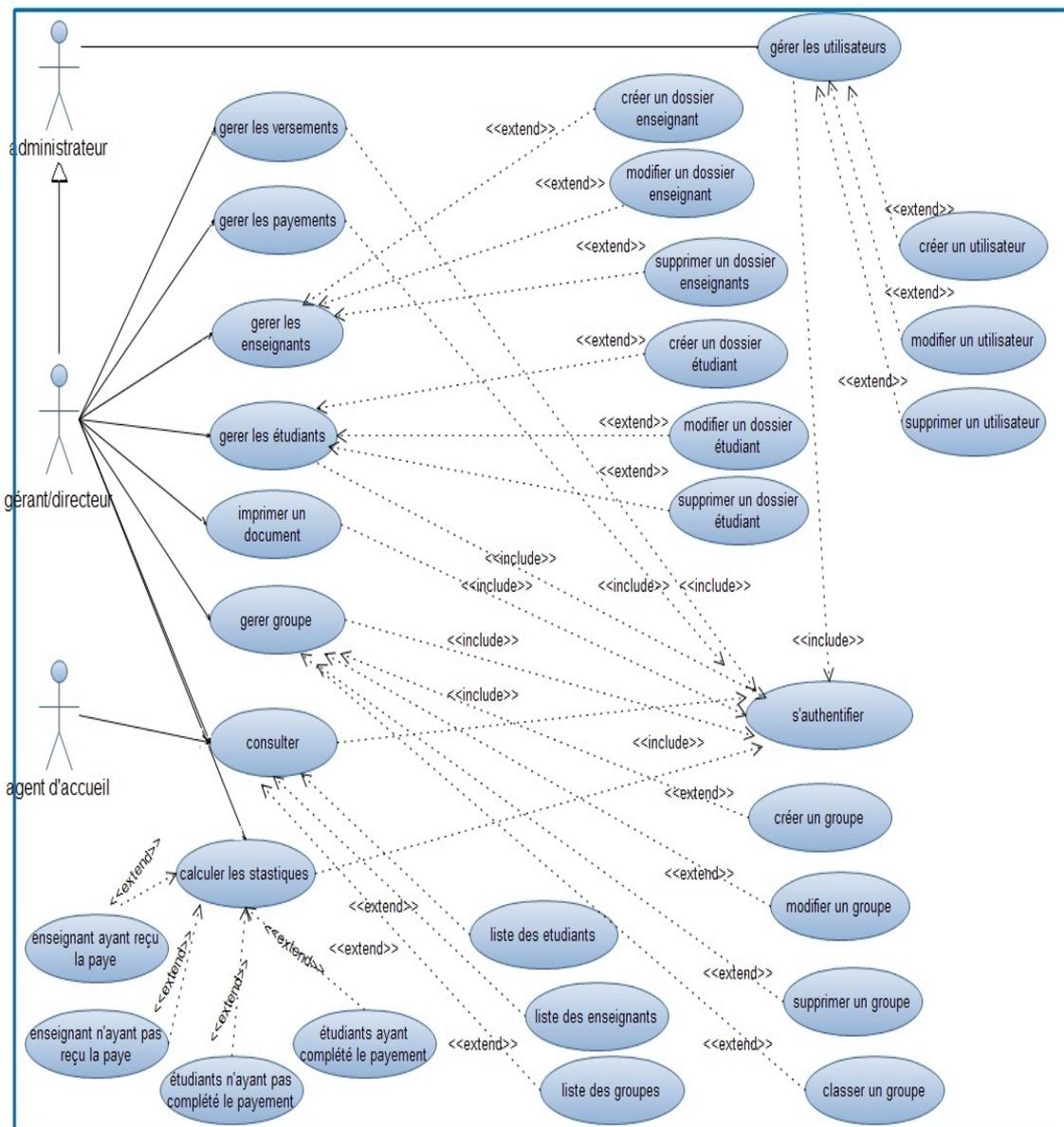


FIGURE 2.1 – Diagramme des cas d'utilisation globale

En s'appuyant sur la figure (2.1), nous avons distingué les cas d'utilisations suivants :

1. Gestion des utilisateurs : pour l'administrateur de site.
2. Authentification : l'application doit vérifier que l'utilisateur est bien celui qui prétend être afin de lui autoriser l'accès à l'application.
3. Mise à jour des données qui comprend : la création, la modification et la suppression d'un étudiant/enseignant/groupe.
4. Imprimer un document : une attestation de formation pour l'étudiant.
5. Consulter un dossier d'étudiant/enseignant : obtenir des informations sur un étudiant (enseignant) quelconque et la liste des groupes.
6. Calcul des statistiques : nombre d'étudiants ayant et n'ayant pas payé leurs formations, le nombre d'enseignants ayant et n'ayant pas reçu une paye.

A l'issue de l'expression des besoins à l'aide du diagramme des cas d'utilisation globale, dans ce qui suit, nous détaillons chacun des cas d'utilisations présenté dans les figure(2.2) jusqu'à la figure (2.11), en donnant sa description textuelle.

A. Le cas d'utilisation s'authentifier

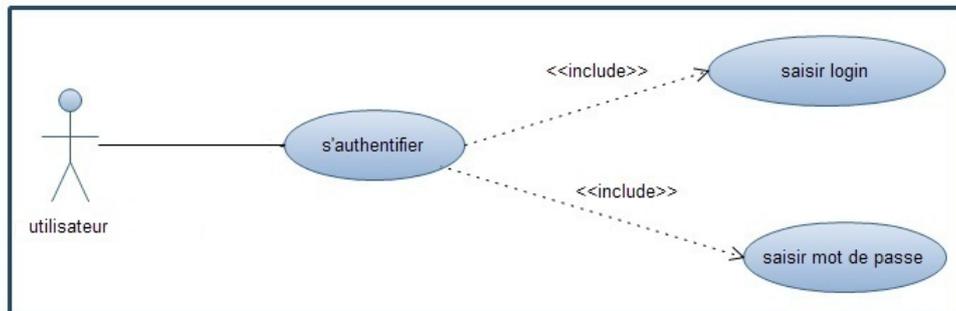


FIGURE 2.2 – Diagramme de cas d'utilisation s'authentifier

- **Acteur principal** :utilisateurs (gérant/directeur, administrateur, agent d'accueil).
- **Objectif** :S'assurer que l'utilisateur est bien celui qui prêtant être.

- **Scenario nominal** :
 1. L'utilisateur saisit son nom d'utilisateur et son mot de passe.
 2. Le système vérifie le nom d'utilisateur et le mot de passe.
 3. Le système affiche l'espace approprié pour chaque utilisateurs.
- **Scenario alternatif** :
 1. Login et mot de passe sont incorrects, un retour vers la page d'authentification sera effectué avec un message d'erreur.

B. Le cas d'utilisation gérer les utilisateurs

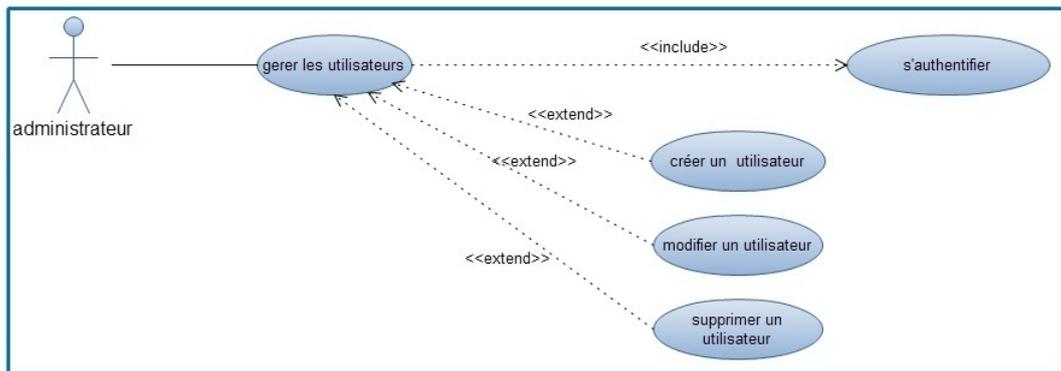


FIGURE 2.3 – Diagramme de cas d'utilisation gérer les utilisateurs

- **Acteur principal** :Administrateur, gérant/directeur.
- **Objectif** :Pouvoir ajouter, modifier, supprimer un dossier utilisateur.
- **Scenario nominal** :

cas 1 : Créer un utilisateur

1. L'administrateur choisit d'ajouter un dossier utilisateur.
2. Le système affiche le formulaire à remplir.
3. L'administrateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des utilisateurs et l'affiche.

cas 2 : Modifier un dossier utilisateur

1. L'administrateur choisit l'utilisateur à modifier.
2. Le système affiche le formulaire de modification.
3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des utilisateurs et l'affiche.

cas 3 : Supprimer un dossier utilisateur

1. L'administrateur choisit l'utilisateur à supprimer.
2. Le système demande une confirmation.
3. L'administrateur confirme ou annule la suppression.
4. Le système supprime l'utilisateur de la base.
5. Le système actualise la liste des utilisateurs et l'affiche.

- **Scenario alternatif** :

cas 1 :

1. Utilisateur existe déjà ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.

cas 2 :

1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.

cas 3 :

1. utilisateur inexistant : un message d'erreur sera affiché.

C. Le cas d'utilisation gérer les étudiants

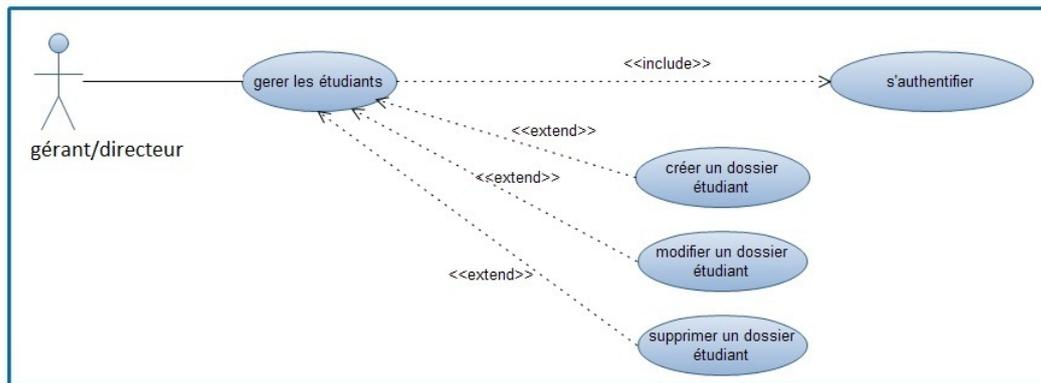


FIGURE 2.4 – Diagramme de cas d'utilisation gérer les étudiants

- **Acteur principal** :gérant/directeur.
- **Objectif** :Pouvoir ajouter, modifier, supprimer un dossier étudiant.
- **Scenario nominal** :

cas 1 : Ajouter un dossier étudiant

1. L'utilisateur choisit d'ajouter un dossier étudiant.
2. Le système affiche le formulaire à remplir.
3. L'utilisateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des étudiants et l'affiche.

cas 2 : Modifier un dossier étudiant

1. L'utilisateur choisit l'étudiant à modifier.
2. Le système affiche le formulaire de modification.
3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des étudiants et l'affiche.

cas 3 : Supprimer un dossier étudiant

1. L'utilisateur choisit l'étudiant à supprimer.
2. Le système demande une confirmation.
3. L'utilisateur confirme ou annule la suppression.
4. Le système supprime l'étudiant de la base.
5. Le système actualise la liste des étudiants et l'affiche.

- **Scenario alternatif** :

cas 1 :

1. Étudiant existe déjà ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.

cas 2 :

1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.

cas 3 :

1. Étudiant inexistant : un message d'erreur sera affiché.

D. Le cas d'utilisation gérer les enseignants

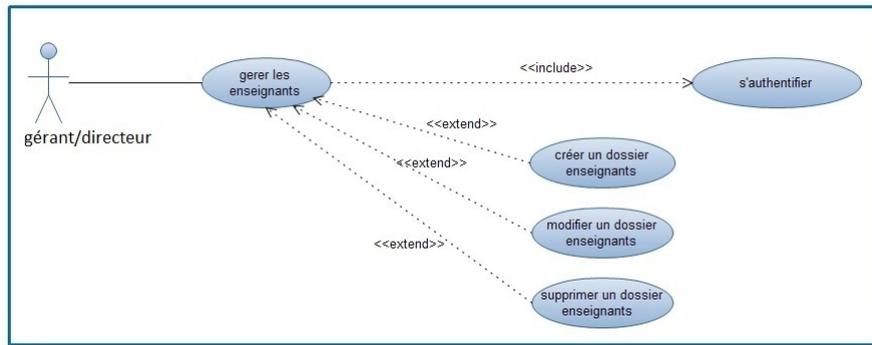


FIGURE 2.5 – Diagramme de cas d'utilisation gérer les enseignants

- **Acteur principal** :gérant/directeur.
- **Objectif** :Pouvoir ajouter, modifier, supprimer un dossier enseignant.
- **Scenario nominal** :

cas 1 : Créer un dossier enseignant

1. L'utilisateur choisit d'ajouter un dossier un enseignant.
2. Le système affiche le formulaire à remplir.
3. L'utilisateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des enseignants et l'affiche.

cas 2 : Modifier un dossier enseignant

1. L'utilisateur choisit l'enseignant à modifier.
2. Le système affiche le formulaire de modification.
3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des enseignants et l'affiche.

cas 3 : Supprimer un dossier enseignant

1. L'utilisateur choisit l'enseignant à supprimer.
2. Le système demande une confirmation.
3. L'utilisateur confirme ou annule la suppression.
4. Le système supprime l'enseignant de la base.
5. Le système actualise la liste des enseignants et l'affiche.

- **Scenario alternatif** :

cas 1 :

1. Enseignant existe déjà ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.

cas 2 :

1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.

cas 3 :

1. Enseignant inexistant : un message d'erreur sera affiché.

E. Le cas d'utilisation gérer les groupes

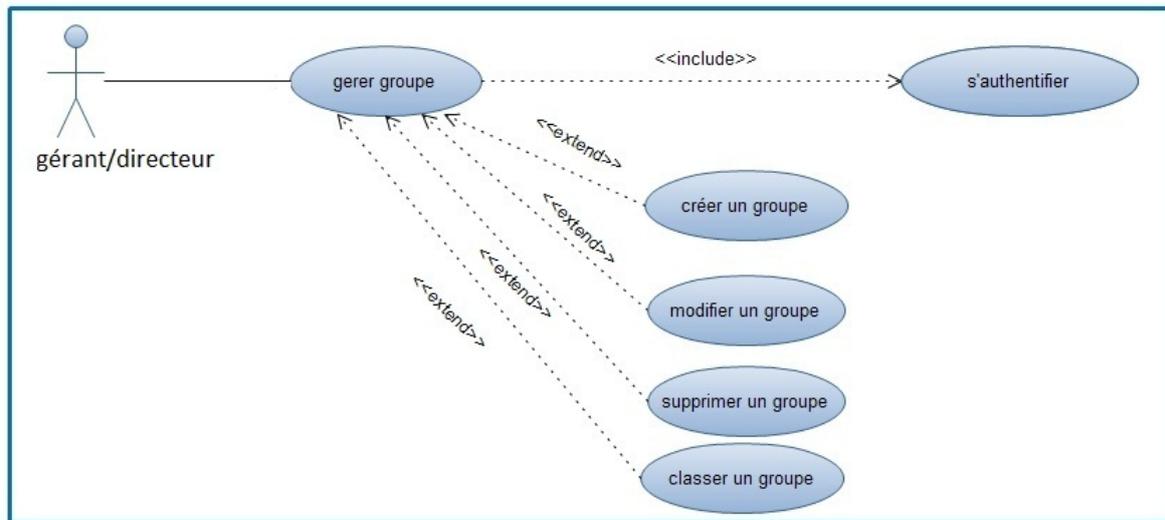


FIGURE 2.6 – Diagramme de cas d'utilisation gérer les groupes

- **Acteur principal** :gérant/directeur.
- **Objectif** :Pouvoir ajouter, modifier, supprimer un groupe.
- **Scenario nominal** :

cas 1 : Créer groupe

1. L'utilisateur choisit d'ajouter un groupe.
2. Le système affiche le formulaire à remplir.
3. L'utilisateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des groupes et l'affiche.

cas 2 : Modifier un groupe

1. L'utilisateur choisit le groupe à modifier.
2. Le système affiche le formulaire de modification.
3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des groupes et l'affiche.

cas 3 : Supprimer un groupe

1. L'utilisateur choisit le groupe à supprimer.
2. Le système demande une confirmation.
3. L'utilisateur confirme ou annule la suppression.
4. Le système supprime le groupe de la base.
5. Le système actualise la liste des groupes et l'affiche.

cas 4 : Classer un groupe

1. L'utilisateur choisit de classer un groupe.
2. Le système affiche le formulaire pour choisir.

3. L'utilisateur choisit soit de lancer le groupe ou de le classer.
4. Le système actualise la liste des groupes et l'affiche.

• **Scenario alternatif :**

cas 1 :

1. Groupe existe déjà ou champs non conforme aux types, formulaire vide : un message d'erreur sera affiché.

cas 2 :

1. Modification avec des champs vides, champs non conforme aux types : un message d'erreur sera affiché.

cas 3 :

1. Groupe inexistant : un message d'erreur sera affiché.

cas 4 :

1. Groupe déjà classé : un message d'erreur sera affiché.

F. Le cas d'utilisation consulter un dossier étudiant

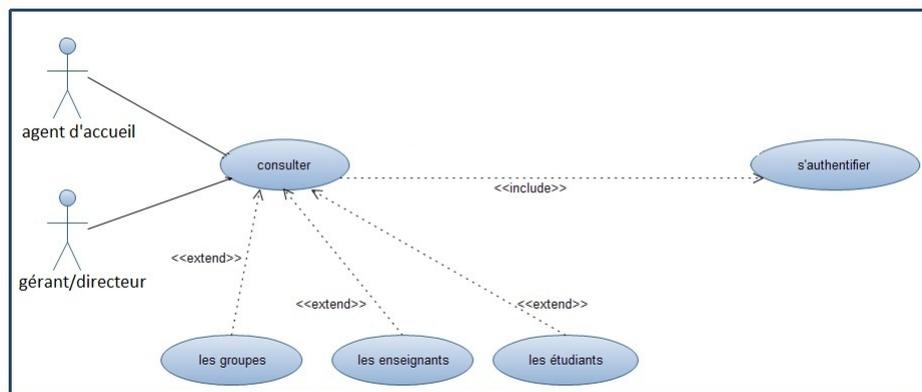


FIGURE 2.7 – Diagramme du cas d'utilisation consulter un dossier étudiant

• **Acteur principal :** gérant/directeur, agent d'accueil.

• **Objectif :** Prendre connaissances de l'étudiant traité et d'informations antérieures relatives à son dossier.

• **Scenario nominal :**

1. L'utilisateur saisit son login et mot de passe.
2. Le système affiche la page d'accueil.
3. L'utilisateur recherche l'étudiant par son numéro identifiant.
4. Le système affiche le formulaire de l'étudiant correspondant.
5. L'utilisateur consulte les informations qu'il juge pertinentes.

• **Scenario alternatif :**

1. Le système n'arrive pas à identifier l'utilisateur et affiche un message d'erreur.
2. Le système ne trouve pas l'étudiant, il demande de ressaisir les informations de nouveau.

Note : c'est le même scénario pour la consultation des enseignants et des groupes.

G. Le cas d'utilisation gérer le paiement des enseignants

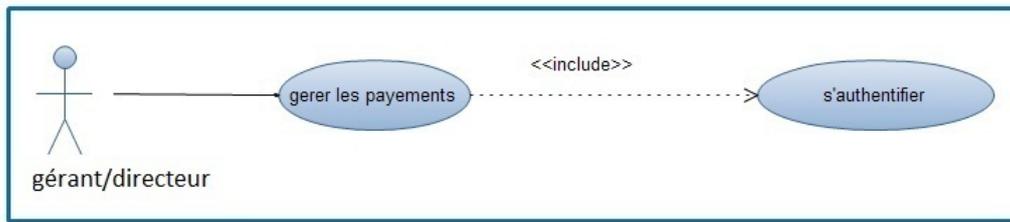


FIGURE 2.8 – Diagramme de cas d'utilisation gérer paiement

- **Acteur principal** :gérant/directeur.
- **Objectif** :avoir une liste des enseignants ayant reçu leur paye ainsi que ceux qui l'ont pas reçu.
- **Scénario nominal** :
 1. Le gérant saisi son login et son mot de passe.
 2. Le système affiche la page d'accueil.
 3. Le gérant choisit d'ajouter la paye ou une tranche de la paye pour l'enseignant.
 4. Le système affiche la liste des paiements effectué.
- **Scénario alternatif** :
 1. Le système n'arrive pas à identifier le gérant et affiche un message d'erreur.

H. Le cas d'utilisation gérer les versements des étudiants

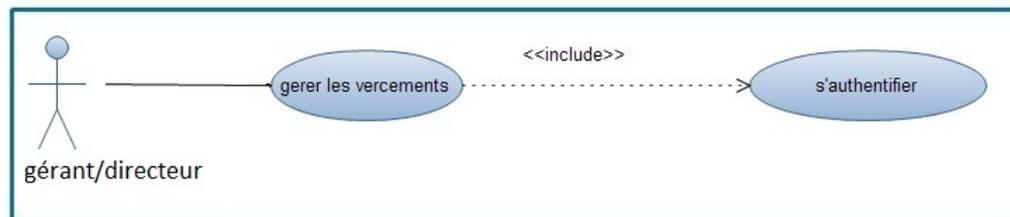


FIGURE 2.9 – Diagramme de cas d'utilisation gérer les versements des étudiants

- **Acteur principal** :gérant/directeur.
- **Objectif** :avoir un historique de tout les paiements effectués par les étudiants.
- **Scénario nominal** :
 1. Le gérant saisi son login et son mot de passe.
 2. Le système affiche la page d'accueil.
 3. Le gérant choisit d'ajouter le versement effectué par l'étudiant
 4. Le système affiche la liste des versements effectué.
- **Scénario alternatif** :
 1. Le système n'arrive pas à identifier le gérant et affiche un message d'erreur.

I. Le cas d'utilisation imprimer un document

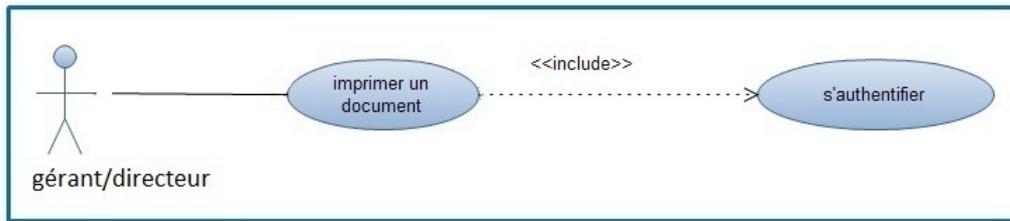


FIGURE 2.10 – Diagramme de cas d'utilisation imprimer un document

- **Acteur principal** :gérant.
- **Objectif** : Imprimer un document (attestation de réussite).
- **Scenario nominal** :
 1. Le gérant saisi son login et son mot de passe.
 2. Le système affiche la page d'accueil.
 3. Le gérant lance l'impression.
 4. Le document demandé est imprimé.
- **Scenario alternatif** :
 1. Le système n'arrive pas à identifier le gérant et affiche un message d'erreur.

J. Le cas d'utilisation calculer les statistiques

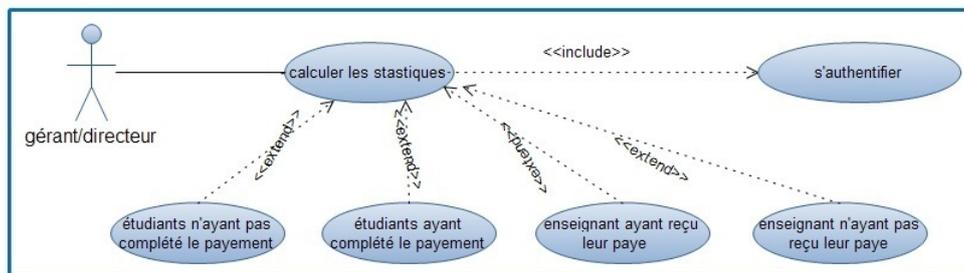


FIGURE 2.11 – Diagramme de cas d'utilisation calculer les statistiques

- **Acteur principal** :gérant/directeur.
- **Objectif** : Calculer les statistiques pour des étudiants ayant payé ou non leurs frais de formations(réspectivement pour les enseignants ayant et n'ayant pas reçu leurs paye).
- **Scenario nominal** :
 1. Le gérant accède à la page d'accueil après authentification.
 2. Le gérant va sur 'calculer les statistiques', et choisi les statistiques qu'il désire savoir.
 3. Le système calcule les statistiques et affiche le résultat.
- **Scenario alternatif** :
 1. Le système ne peut identifier le gérant et demande de ressaisir son login et mot de passe une autre fois.

Après avoir vue les différent diagrammes des cas d'utilisation, juste après et en vue de mettre en évidence les objets métiers impliqués dans les interactions utilisateur-système nous allons exprimer quelques cas d'utilisations à l'aide des diagrammes des séquences.

2.6 Le diagramme des séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Ces communications entre les classes sont reconnues comme des messages. Le diagramme des séquences énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps [21].

Pour réaliser les diagrammes des séquences nous avons utilisé des opérateurs d'interactions.

Un opérateur d'interaction définit le type d'un fragment composé.

Les opérateurs d'interaction que nous avons utilisés dans les diagrammes de séquences sont :

1. Référence (ref) : cet opérateur désigne que le fragment fait référence à un cas vue précédemment.
2. Alternative(Alt) :cet opérateur désigne que le fragment composé représente un choix de comportement. Une opérande d'interaction au maximum sera choisie. L'opérande choisi doit avoir une expression de garde implicite ou explicite qui a la valeur "true" à ce point de l'interaction.
3. Loop : cet opérateur désigne que le fragment composé représente une boucle. L'opérande "loop" sera répétée plusieurs fois [13].

Les figures (2.12) jusqu'à (2.17) présenteront respectivement les diagrammes des séquences : s'authentifier, ajouter/modifier/supprimer consulter un dossier étudiant et imprimer un document.

A. Diagramme de séquence s'authentifier

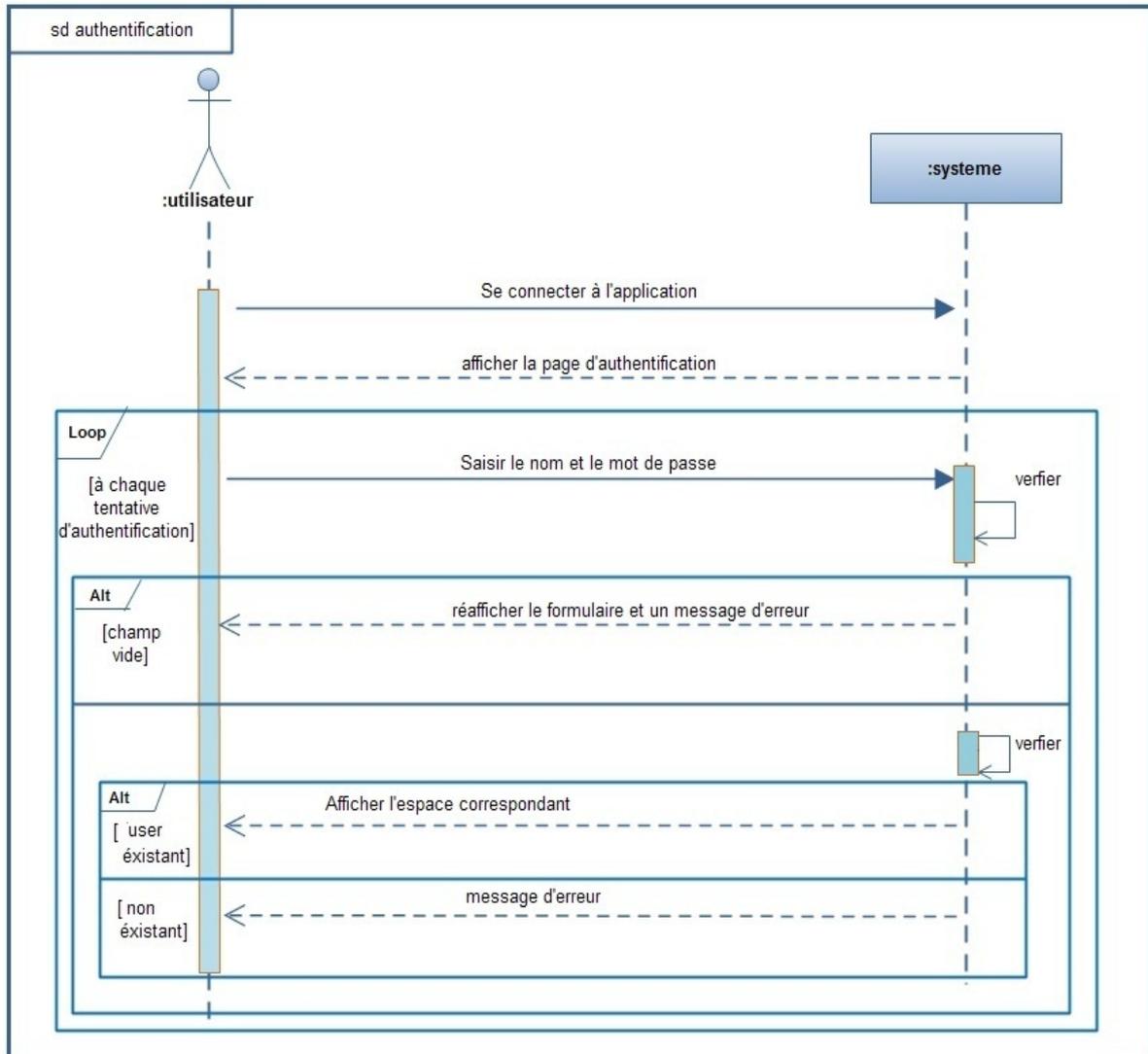


FIGURE 2.12 – Diagramme séquence système lié au cas d'utilisation ”s'authentifier”

B. Diagramme de séquence ajouter un dossier étudiant

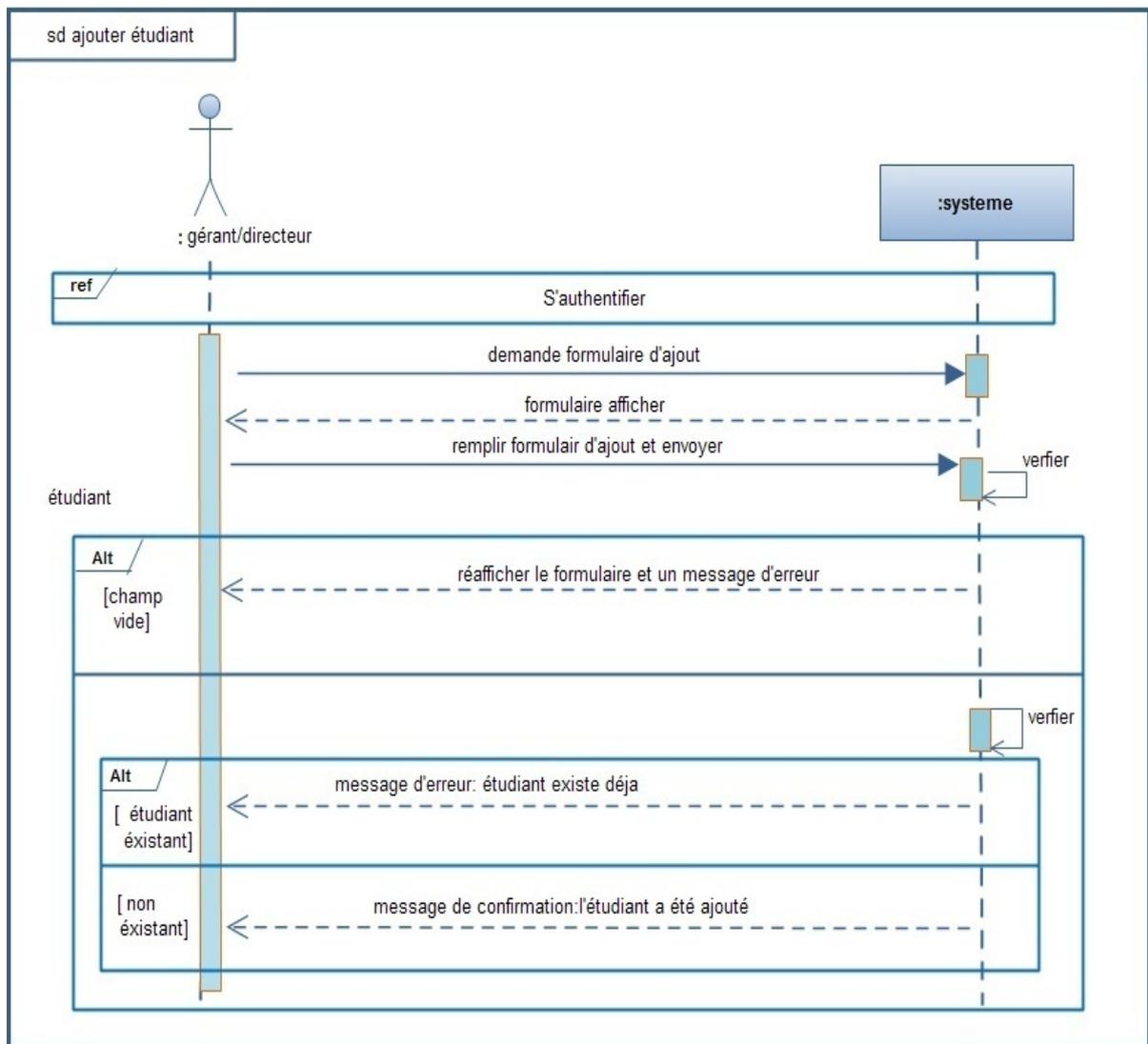


FIGURE 2.13 – Diagramme séquence système lié au cas d'utilisation "ajouter un dossier étudiant"

C. Diagramme de séquence modifier un dossier étudiant

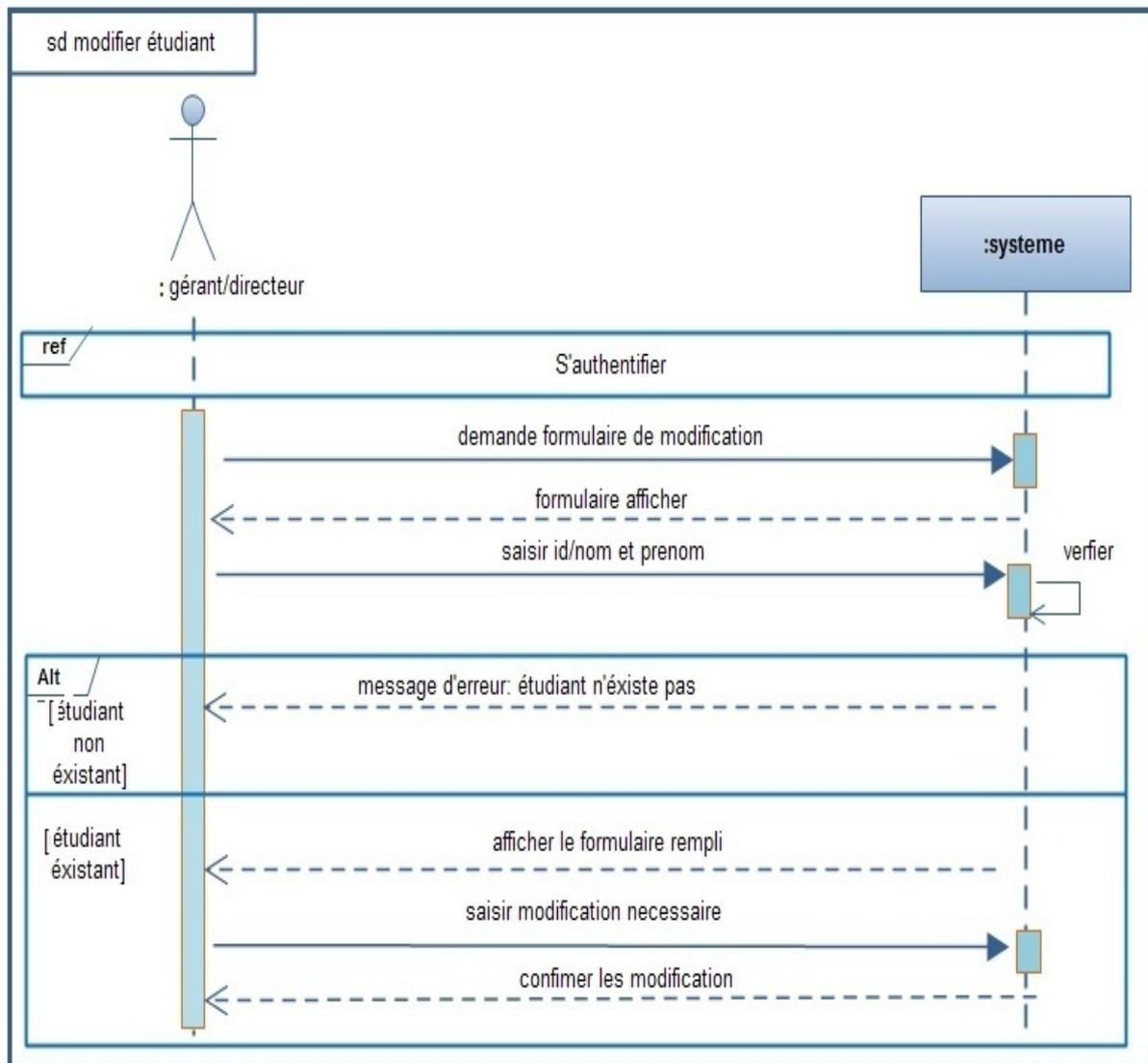


FIGURE 2.14 – Diagramme séquence système lié au cas d'utilisation "modifier un dossier étudiant"

D. Diagramme de séquence supprimer un dossier étudiant

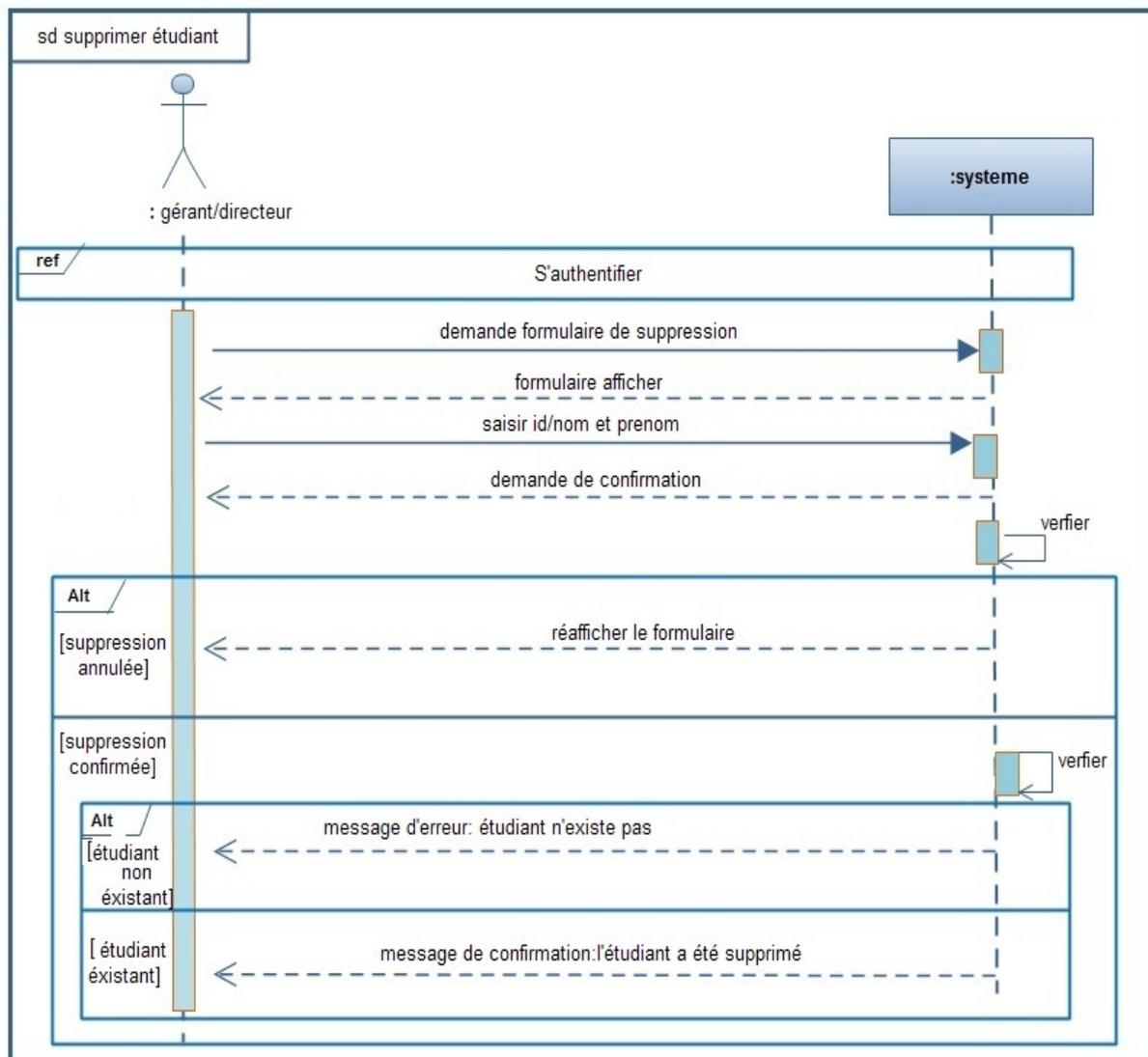


FIGURE 2.15 – Diagramme séquence système lié au cas d’utilisation ”supprimer un dossier étudiant”

Note : c’est les même étapes pour les diagrammes de séquences de la gestion des enseignant et des groupes.

E. Diagramme de séquence Consulter un étudiant

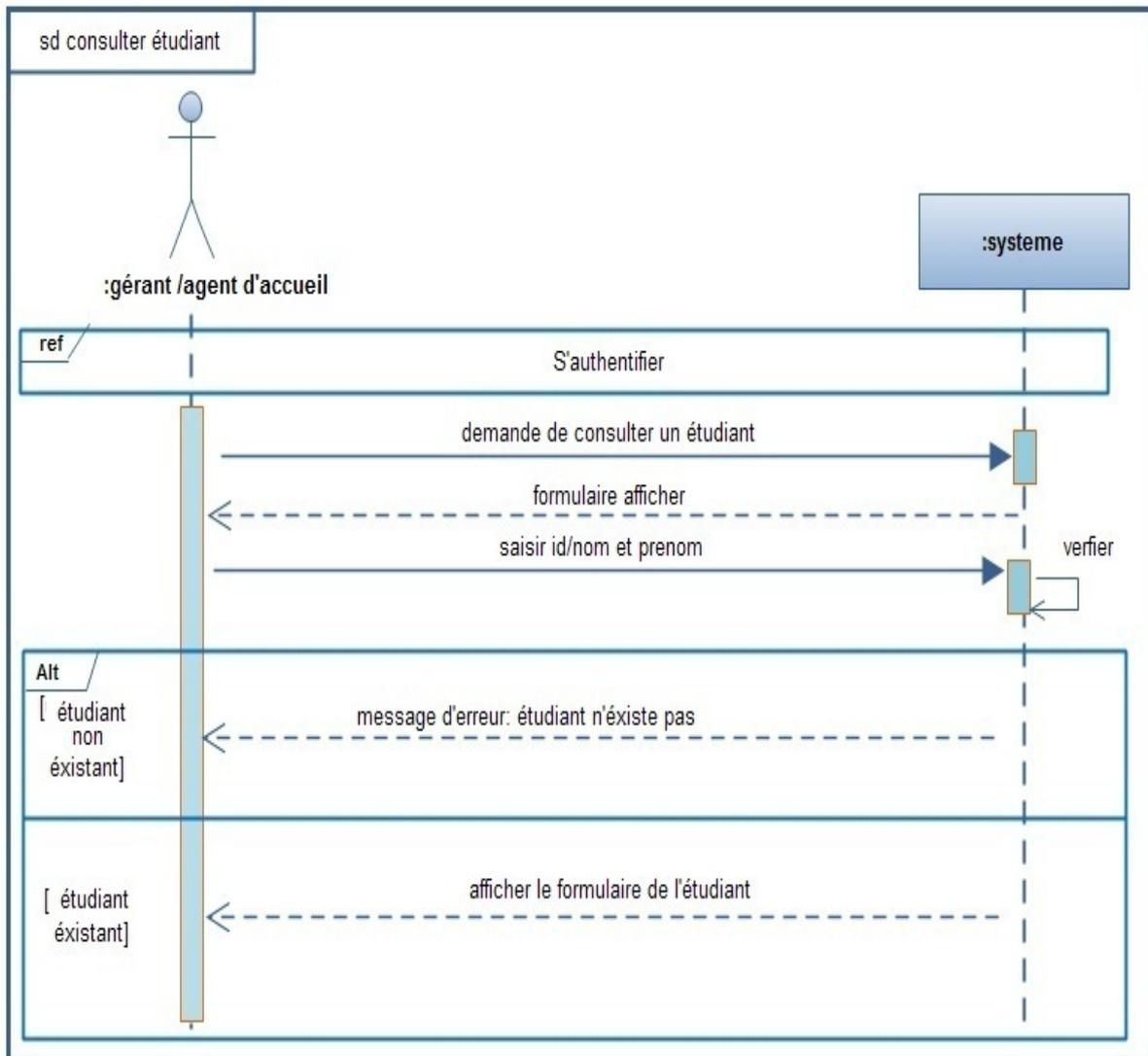


FIGURE 2.16 – Diagramme séquence système lié au cas d'utilisation "consulter un étudiant"

F. Diagramme de séquence imprimer un document

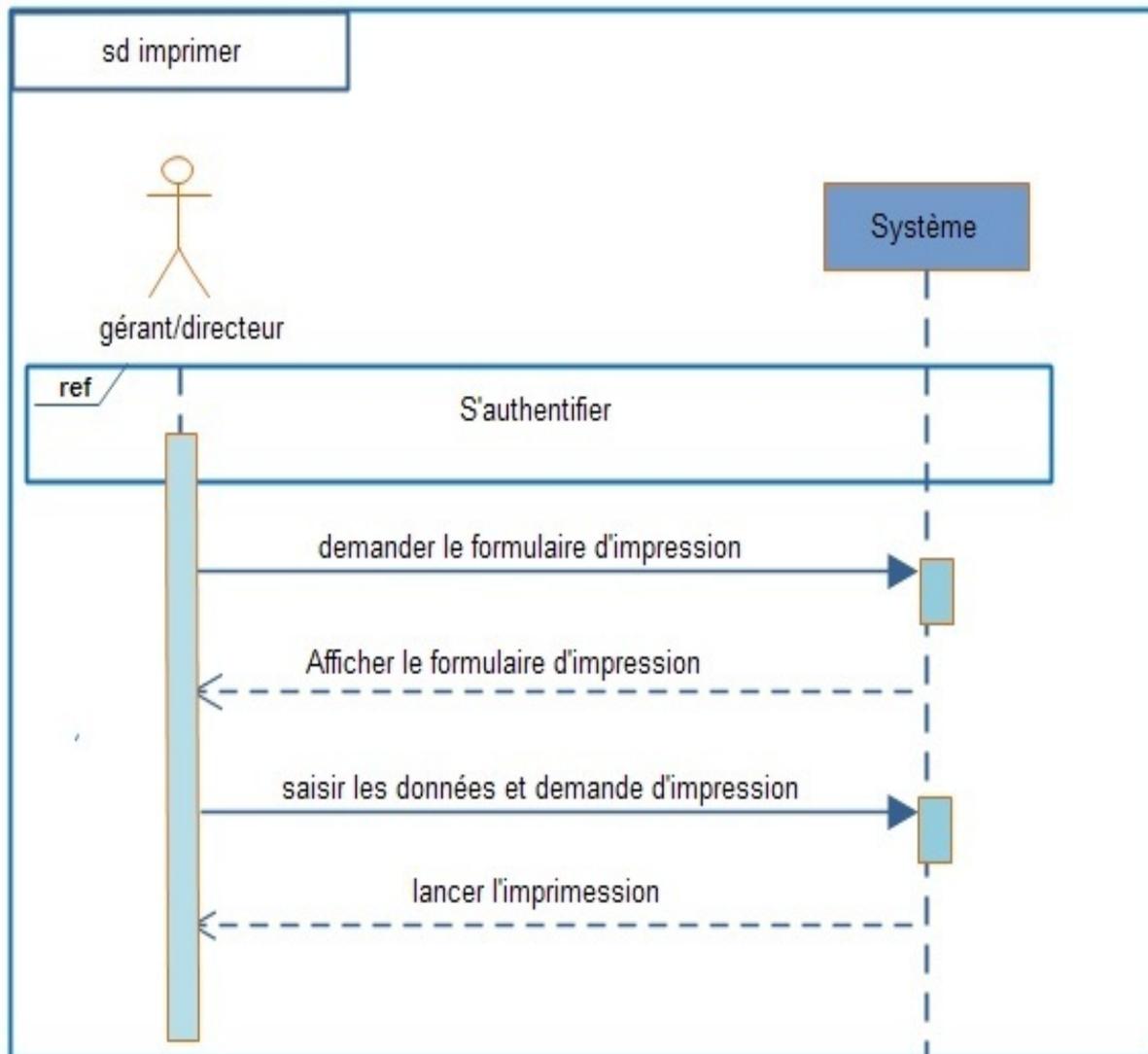


FIGURE 2.17 – Diagramme séquence système lié au cas d'utilisation "imprimer un document"

2.7 Conclusion

Dans la première partie de ce chapitre, nous avons décrit le contexte de notre projet puis nous avons posé la problématique et tracé quelques objectifs.

Dans la deuxième partie, nous nous sommes intéressé à analyser les besoins de notre application et à spécifier les différents cas d'utilisations en abordant le diagramme des cas d'utilisations et les diagrammes des séquences.

Dans ce qui suit nous allons entamer la conception en présentant les diagrammes des interactions et le diagramme des classes.

Conception

3.1 Introduction

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité.

Dans ce qui suit nous présentons les diagrammes d'interactions, puis nous passerons à la description des classes participantes dans le diagramme des classes, et nous construirons le modèle relationnel à partir des classes de l'application, et sa en suivant les règles du passage au relationnel.

3.2 Diagramme d'interaction

Le diagramme d'interaction est un diagramme de séquence détaillé.

Nous présentons dans cette section les diagrammes d'interactions, en remplaçant le système représenté par une boîte noire dans le diagramme de séquences par les trois types de classes suivantes :

1. L'objet interface (dialogue)(<D>) : représente l'interface entre l'acteur et le système,
2. L'objet contrôle (<Ctrl>) :représente un traitement du système déclenché par un acteur,
3. L'objet entité (<E>) :représente des objets décrits dans le cas d'utilisation[29].

Nous respecterons également les règles que nous avons fixées sur les relations entre les classes d'analyses, mais en nous intéressant cette fois-ci aux interactions dynamiques entre objets :

- Les acteurs ne peuvent interagir (envoyer des messages) qu'avec les dialogues ;
- Les dialogues peuvent interagir avec les contrôles ;
- Les contrôles peuvent interagir avec les dialogues, les entités, ou d'autres contrôles ;
- Les entités ne peuvent interagir qu'entre elles.

Les figures (3.1), (3.2), (3.3), (3.4) sises ci-dessous, présenterons respectivement les diagrammes d'interactions s'authentifier, ajouter, modifier et supprimer un dossier étudiant.

3.2.1 Diagramme d'interaction authentification d'un utilisateur

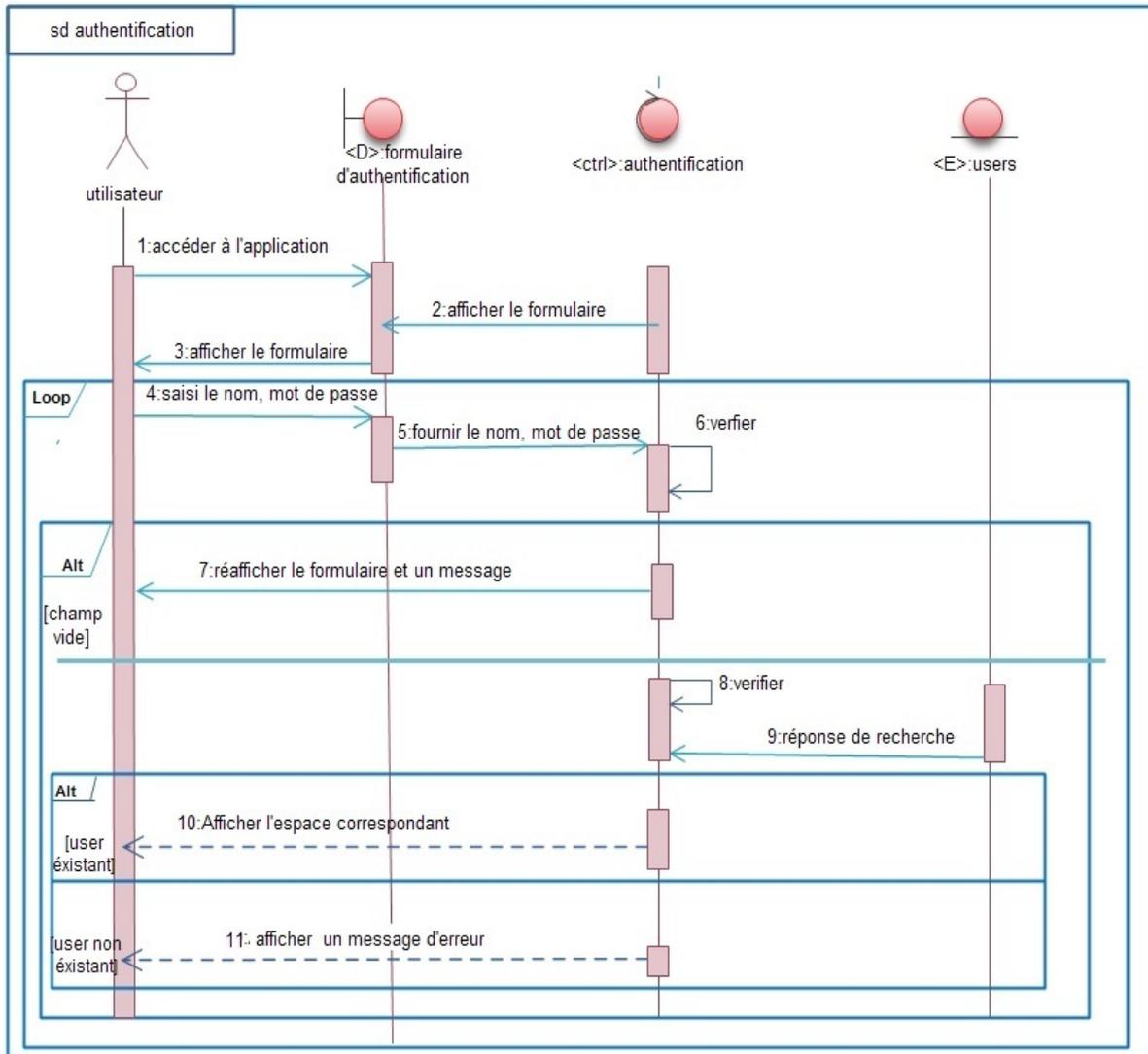


FIGURE 3.1 – Diagramme d'interaction authentification d'un utilisateur

3.2.2 Diagramme d'interaction gérer un dossier étudiant

cas : ajouter un dossier étudiant

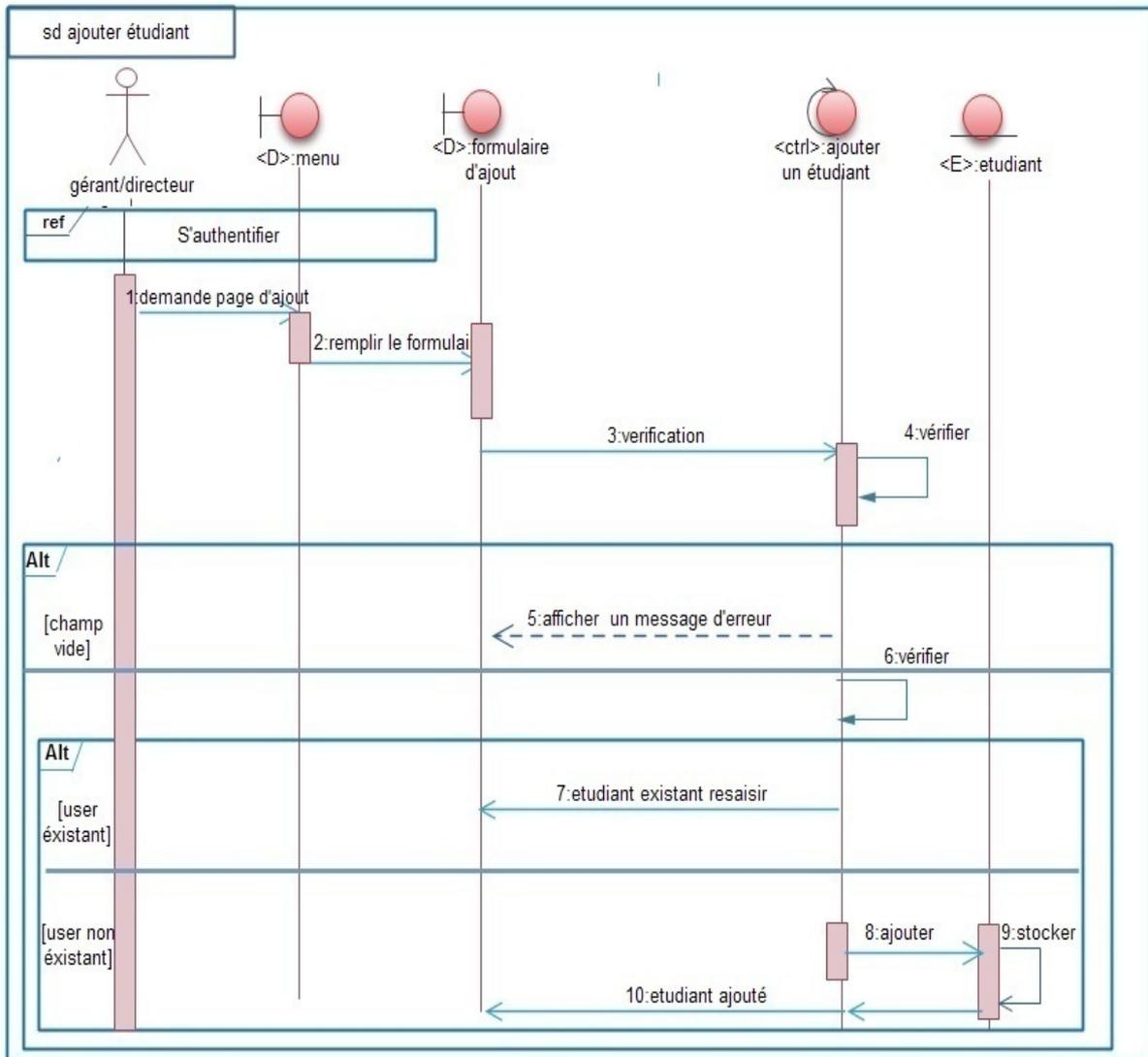


FIGURE 3.2 – Diagramme d'interaction ajouter un dossier étudiant

cas : modifier dossier un étudiant

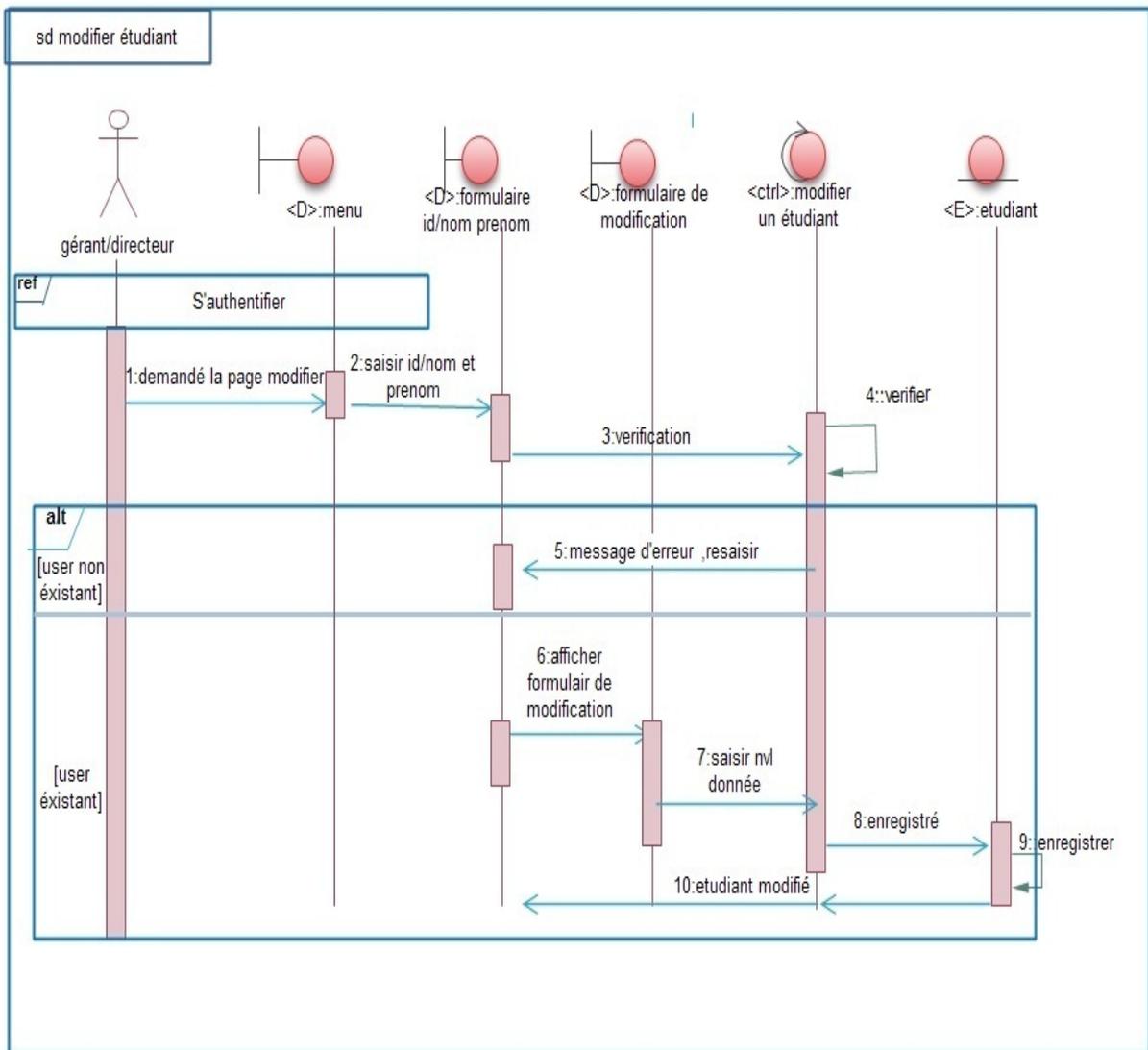


FIGURE 3.3 – Diagramme d'interaction modifier un dossier étudiant

cas : supprimer un dossier étudiant

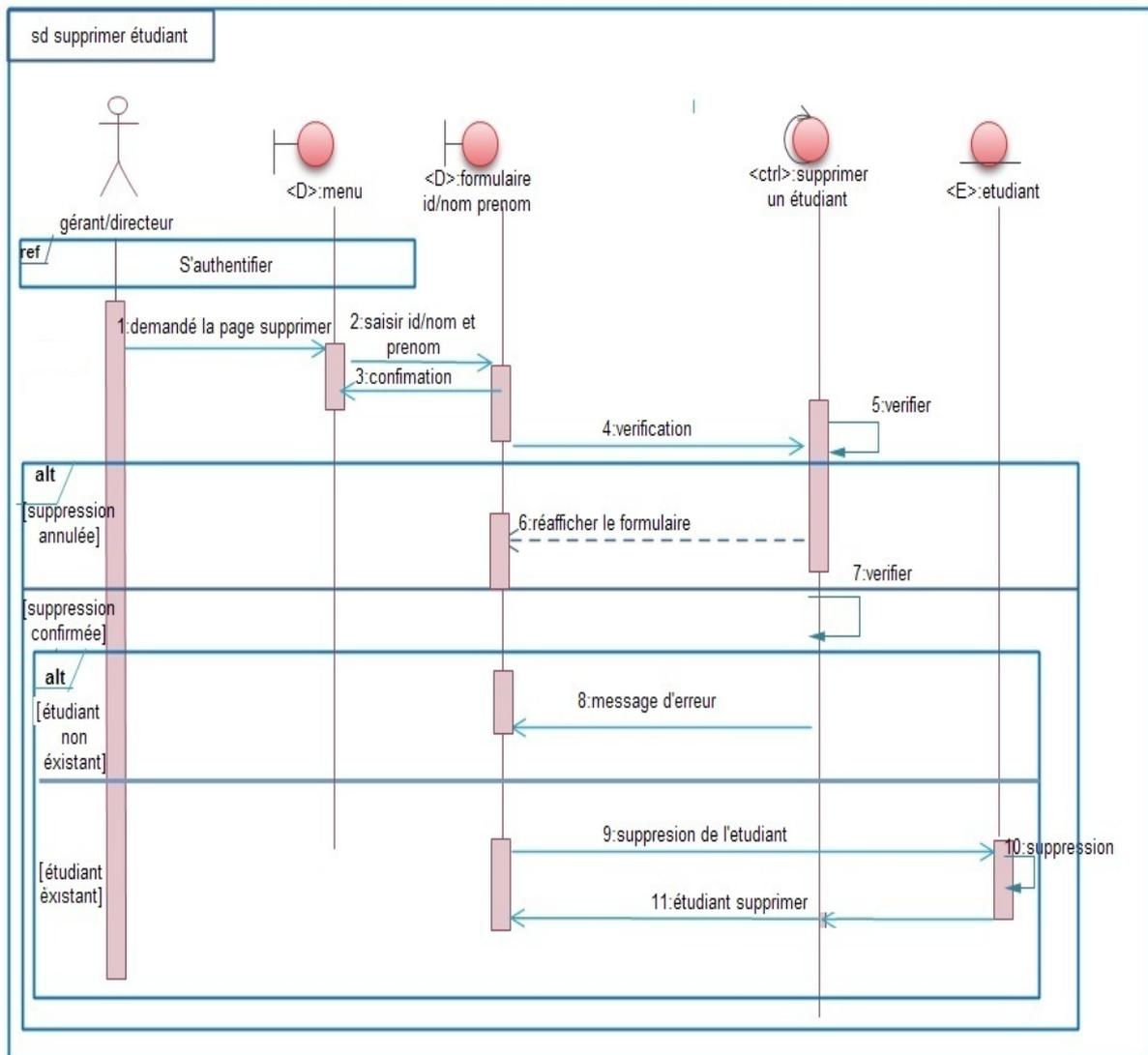


FIGURE 3.4 – Diagramme d’interaction supprimer un dossier étudiant

Note : c’est les même étapes pour les diagrammes d’interactions de la gestion des enseignant et des groupes.

Après avoir spécifié les cas d’utilisations de notre application, il nous reste à décrire les classes participantes dans notre application et à présenter notre diagramme des classes.

3.3 Diagramme des classes

Un diagramme des classes décrit le type des objets ou données du système ainsi que les différentes formes de relations statiques qui relient entre eux [21].

3.3.1 Description des classes

Étudiant : c'est l'entité essentielle de l'application, il passe d'abord par le gérant ou l'agent d'accueil et effectue une inscription pour une formation.

Enseignant : il est responsable sur les groupes des étudiants, et reçoit une paye par le gérant.

Groupe : contient des étudiant qui reçoivent des formations, chaque groupe est pris en charge par un enseignant.

Users : – gérant : il se charge de suivre l'étudiant/enseignants/groupes (faire une inscription, choisir une formation, payer les frais de formation...).

– agent d'accueil : c'est la personne qui est chargée de consulter les dossiers des étudiants/enseignants/groupes.

Versement : les étudiants font des versements pour la formation.

Paiement : les enseignants reçoivent des payes.

La description de ces classes est donnée par le tableau (3.1) suivant :

Classes	Attribut mots	Méthodes
Etudiant	Id-etud, nom, Prenom, date/lieu de naiss, statut, niveau scolaire, Adresse, Tel, note examen, groupe, versement	Ajouter(), Modifier(), Supprimer(), Consulter()
Enseignant	id-ens, nom, prénom, adresse, tel, email, langue, rémunéré, versement	Ajouter(), Modifier(), Supprimer(), Consulter()
Groupe	id-groupe, nom, langue, niveau, etat	Ajouter(), Modifier(), Supprimer(), Classer()
Versement	id-vers, montant, reste, date	ajouté()
paiement	id-pay, montant, reste, date	ajouté()
Users	Id-user, User, Password	Creer(), Modifier(), Supprimer()

TABLE 3.1 – Description des classes de l'application

Après la description des classes présentées dans le tableau (3.1), la figure (3.5) présentera le diagramme des classes de notre application :

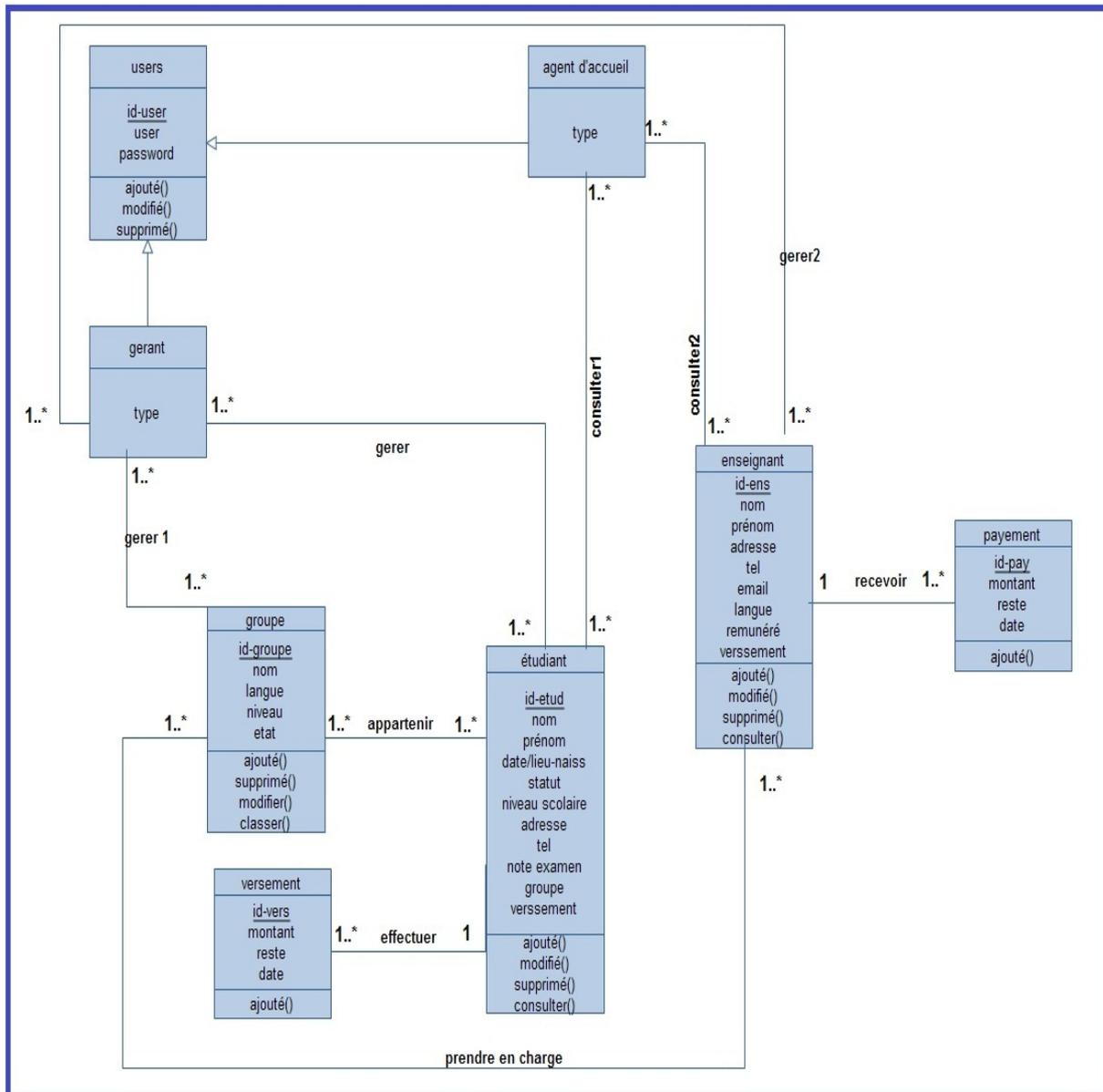


FIGURE 3.5 – Diagramme des classes

Le dictionnaire des données de tables présentées dans le diagramme des classes est donné par le tableau (3.2) suivant :

Classe Etudiant		
Attribut	Description	Type
Id-etud	Identifiant de l'étudiant	entier
nom	Nom de l'étudiant	varchar(100)
Prenom	Prénom de l'étudiant	varchar(100)
date	Date de naissance de l'étudiant	Date (jj/mm/aa)
lieu	lieu de naissance de l'étudiant	Varchar(100)
statut	Statut de l'étudiant (étudiant, enfant, fonctionnaire)	Varchar(100)
niveau scolaire	Niveau scolaire de l'étudiant	Varchar(100)
Adresse	Adresse de l'étudiant	Varchar(100)
Tel	Téléphone de l'étudiant	entier
note examen	Note obtenue lors du test d'examen de début de formation	Varchar(100)
groupe	Le groupe ou l'étudiant appartient	Varchar(100)
versement	Les frais de la formation en tranche	entier
Classe Enseignant		
Attribut	Description	Type
id-ens	Identifiant de l'enseignant	entier
nom	Nom de l'enseignant	Varchar(100)
Prenom	Prénom de l'enseignant	Varchar(100)
adresse	Adresse de l'enseignant	Varchar(100)
tel	Numéro de téléphone de l'enseignant	entier
email	Adresse email de l'enseignant	Varchar(100)
langue	Langue enseignée par l'enseignant	Varchar(100)
remunéré	Salaire de l'enseignant	entier
versement	Salaire versé par tranche	entier

Classe Groupe		
Attribut	Description	Type
id-groupe	Identifiant du groupe	entier
nom	Nom du groupe	Varchar(100)
langue	Langue enseigné dans ce groupe	Varchar(100)
niveau	Niveau concerne la langue	Varchar(100)
etat	Etat du groupe (à lancé, lancé, classé)	Varchar(100)
Classe Versement		
Attribut	Description	Type
id-vers	Identifiant du versement	entier
id-etud	Identifiant de l'étudiant qui a verser	entier
montant	Le montant à verser	entier
reste	Le montant qui reste à verser	entier
date	Date du versement	Date (jj/mm/aa)
Classe Payment		
Attribut	Description	Type
id-pay	Identifiant du paiement	entier
id-ens	Identifiant de l'enseignant à payer	entier
montant	Le montant à payer	entier
reste	Le montant qui reste à payer	entier
date	Date du paiement	Date (jj/mm/aa)
Classe Users		
Attribut	Description	Type
id-user	Identifiant de l'utilisateur	entier
User	Login de l'utilisateur (gérant, agent d'accueil)	Varchar(100)
Password	Le mot de passe de l'utilisateur	Varchar(100)

TABLE 3.2 – Dictionnaire des données

Après avoir vue le diagramme des classes, nous passerons maintenant au modèle relationnel.

3.4 Passage au relationnel

3.4.1 Règles de passage du diagramme de classes au modèle relationnel

Le passage du diagramme de classes au modèle relationnel ne se fait pas au hasard. Il existe un certain nombre de règles qui nous permettent de réaliser cette opération [28].

- **Règle 1 : Transformation des classes** Chaque classe devient une relation. L’identifiant (respectivement les attributs) de la classe devient la clé primaire (respectivement des attribut) de la relation.
- **Règle 2 : Association un-à-plusieurs** Il faut ajouter un attribut de type clé étrangère dans la relation *fil*s de l’association.
L’attribut porte le nom de la clé primaire de la relation *père* de l’association.
- **Règle 3 : Association plusieurs-à-plusieurs** l’association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l’association.
- **Règle 4 : Association un-à-un** Il faut ajouter un attribut clé étrangère, dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L’attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l’association.
- **Règle 5 : Transformation de l’héritage** Trois décompositions sont possibles pour traduire une association d’héritage en fonction des contraintes existantes :
 1. Décomposition par distinction : Il faut transformer chaque sous-classe en une relation. la clé primaire de la surclasse, migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à le fois clé primaire et clé étrangère.
 2. Décomposition descendante : S’il existe une contrainte de totalité ou de partition sur l’association d’héritage, il est possible de ne pas traduire la relation issue de la surclasse.
Il faut alors faire migrer tout ses attributs dans la(les) relation(s) issue(s) de la (des) sous-classe(s).
 3. Décomposition ascendante : Il faut supprimer la relation issue de la sous-classe et faire migrer les attributs dans la relation issue de la surclasse.

Le modèle relationnel doit refléter le diagramme de classes issu de l’analyse, et donc les éléments présents dans le diagramme des classes (classe, attribut, association, classe association) doivent se retrouver dans le modèle relationnel [28].

En procédant par l’application de ces règles, nous avons généré les règles de gestion suivantes :

- Un étudiant appartient à un ou plusieurs groupes.
- Un étudiant fait un ou plusieurs versements.
- Un enseignant prend en charge un ou plusieurs groupes.
- Un enseignant reçoit une ou plusieurs payes.
- Un gérant gère un ou plusieurs étudiants/enseignants/groupes.
- Un agent d’accueil consulte un ou plusieurs étudiants/enseignants.
- Un groupe est pris en charge par un ou plusieurs enseignants.
- Un versement est effectué par un et un seul étudiant.

– Un paiement est reçu par un et un seul enseignant.

En pratiquant les règles de gestion sises ci-dessus nous avons généré le modèle relationnel suivant :

Note : Nous avons utilisé le caractère # pour désigner les clés étrangères, et le soulignement pour désigner les clés primaires.

gérant(id-user, User, Password, id-user-g, User-g, Password-g, type)

agent-d'accueil(id-user, User, Password, id-user-ac, User-ac, Password-ac, type)

Enseignant(id-ens, nom, prénom, adresse, tel, email, langue, rémunéré, versement)

Etudiant(id-etud, nom, Prenom, date/lieu de naiss, statut, niveau scolaire, Adresse, Tel, note examen, groupe, versement)

Groupe(id-groupe, nom, langue, niveau, etat)

gérer(#id-user-g, #id-etud)

gérer1(#id-user-g, #id-ens)

gérer2(#id-user-g, #id-groupe)

consulter1(#id-user-ac, #id-etud)

consulter2(#id-user-ac, #id-ens)

Appartenir(#id-etu, #id-groupe)

prndre-en-charge(#id-ens, #id-groupe)

Versement(id-vers, montant, reste, date, #id-etud)

Paiement(id-pay, montant, reste, date, #id-ens)

3.5 Conclusion

Après avoir abordé les différents diagrammes, on a pu déterminer bien précisément le fonctionnement de notre application web et mettre en évidence l'interaction entre nos acteurs et l'application web.

En effet l'objectif de cette analyse et conception, c'est de concrétiser le système d'information de l'école privée ISA School par la phase implémentation qu'on va aborder dans le chapitre qui va suivre.

Réalisation

4.1 Introduction

Depuis le début du mémoire, nous avons bien déterminé les perspectives de l'application en traçant un cahier de charge (pour plus de précision veuillez consulter l'annexe (a)) qui nous a permis d'avoir un objectif bien concret. Puis en arrivant à l'analyse, nous avons pu avoir une idée bien claire sur comment va être notre applications web, et c'est ce qui nous a mené à mieux comprendre ses fonctionnalités.

Nous arrivons maintenant à la dernière phase qui est l'implémentation.

4.2 Environnement et outils de développement de notre application

Pour pouvoir bien réaliser notre application nous avons opté pour quelque outils que nous allons définir ci-dessous.

a.PHP (Hypertexte Préprocesseur ou Personal Home Page)

Comme nous l'avant vue déjà précédemment PHP est un langage de scripts libre principalement utilisé pour produire des pages web dynamiques via un serveur http.

b.CSS (CASCADING STYLE SHEETS) [12]

Les feuilles de style dites CSS permettent de gérer la présentation d'une page web. CSS permet de créer des règles de styles et de mise en pages.

Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacement, les bordures, les images de fond, etc.

Le but de CSS est de séparer la structure d'un document HTML et sa présentation.

c.XHTML (Extensible HyperText Markup Language) [18]

XHTML est une combinaison de HTML et de XML (eXtensible Markup Language), est un langage de balisage servant à écrire des pages pour le Word Wide Web(www).

d. JAVASCRIPT [18]

C'est un langage de programmation qui offre la possibilité d'implémenter des traitements élaborés dans des pages web, et permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes de la cote client (-c'est-a-dire au niveau du navigateur et non du serveur web).

Ainsi le langage JavaScript est fortement dépendant du navigateur appelant la page web laquelle le script est incorpore, mais en contrepartie il ne nécessite pas de compilateur.

L'application a été développée à l'aide d'une plate-forme de développement web sous windows qui est le paquetage Wampserver (2.2) pour :

- **Windows** :Un outil qui s'exécute sous Windows xp, Windows 7.
- **Apache (2.2.21)** :c'est un serveur web. Il s'agit du plus important de tous les programmes, il est chargé de délivrer les pages web aux visiteurs. Cependant, Apache ne gère que les sites web statiques. Il faut donc le compléter avec d'autres programmes.
- **MYSQL (5.5.16)** :c'est le système de gestion de base de données (SGBD).
- **PHP(5.3.8)** :c'est un plug-in pour Apache qui le rend capable de traiter des pages web dynamiques[11].

Le schéma illustré dans la figure (4.2) explique le fonctionnement de ces outils dans le système :

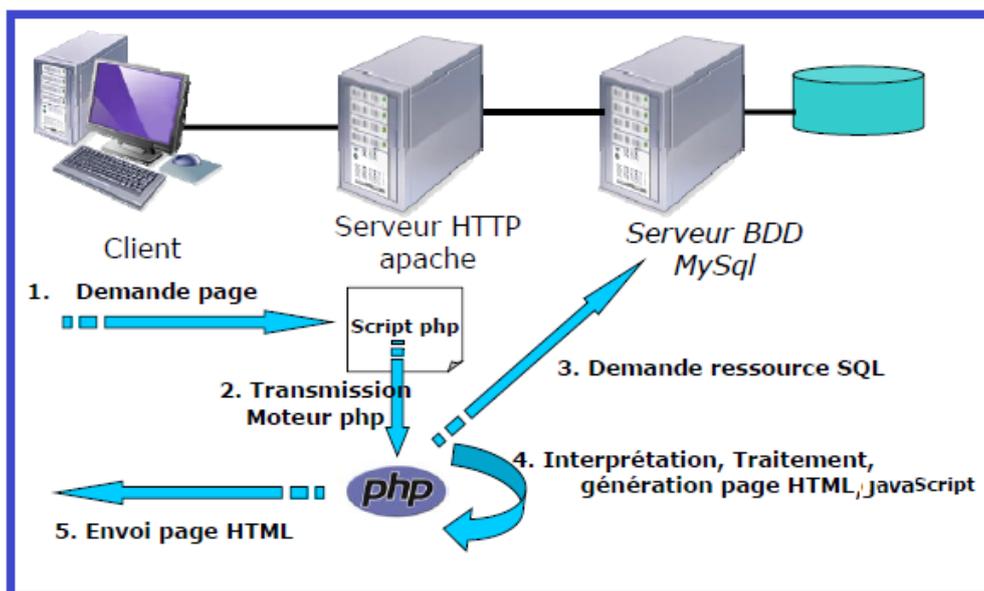


FIGURE 4.1 – Etapes d'une demande de page HTML doté d'un code JavaScript

4.3 Structure et maquette de l'application

Le schéma ci-après, présenté dans la figure (4.2) récapitule l'organisation des pages de l'application (Diagramme de navigation)

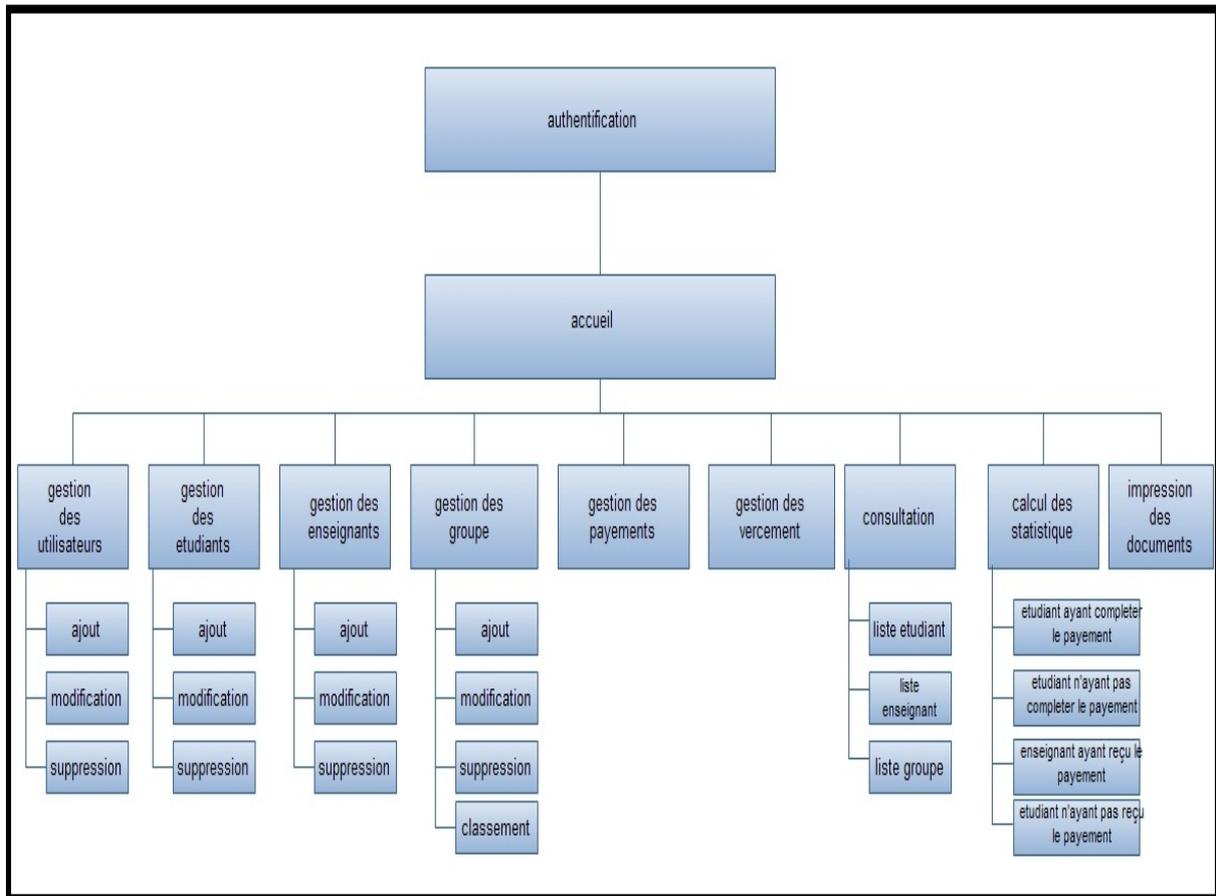


FIGURE 4.2 – Structure de l'application

Par ailleurs, nous avons aussi procédé après la spécification des besoins vue précédemment et en suivant la structure présentée dans la figure (4.2) un ensemble de maquettes élaborés pour les utilisateurs potentiels de l'application. Les figures (4.3) et (4.4) suivantes illustrent quelques une :

4.3.1 Maquette de la page d'accueil après l'authentification

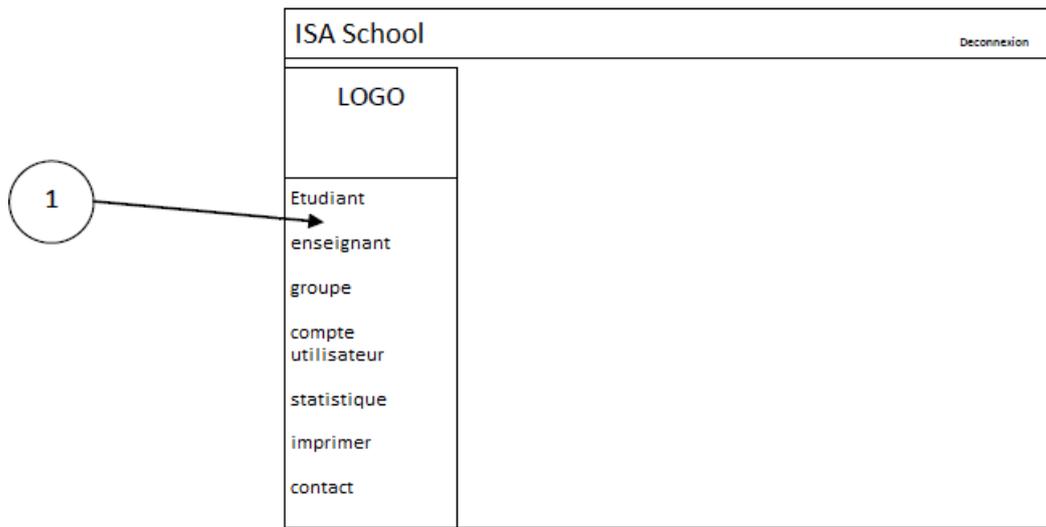


FIGURE 4.3 – maquette de la page d'accueil

1- Différents rubriques d'application correspondantes aux fonctionnalités.

4.3.2 Maquette de la page d'ajout d'un étudiant

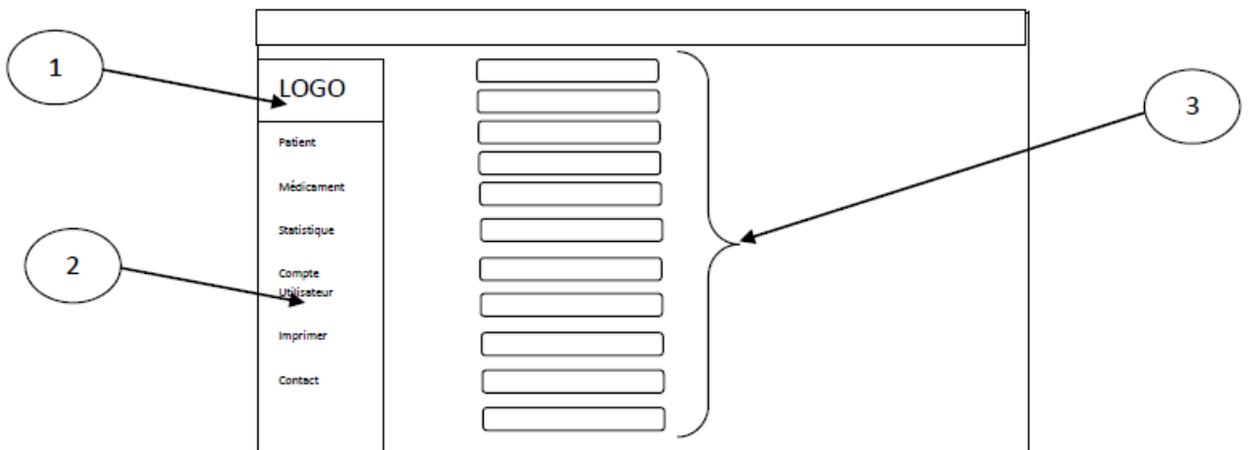


FIGURE 4.4 – maquette d'ajout d'un étudiant

- 1- Logo de ISA School.
- 2- Différentes rubriques de l'application.
- 3- Champs à remplir pour l'ajout d'un étudiant.

Ainsi de suite, nous avons procédé de la même chose pour les autres maquettes.

4.4 Présentation de quelques interfaces de l'application

A l'aide des maquettes présentées précédemment nous avons abouti à la fin de la construction de notre application.

Les figures (4.5) jusqu'à (4.12) illustrent des exemples de capture d'écrans des différentes interfaces de l'application.

4.4.1 Capture d'écran : s'authentifier

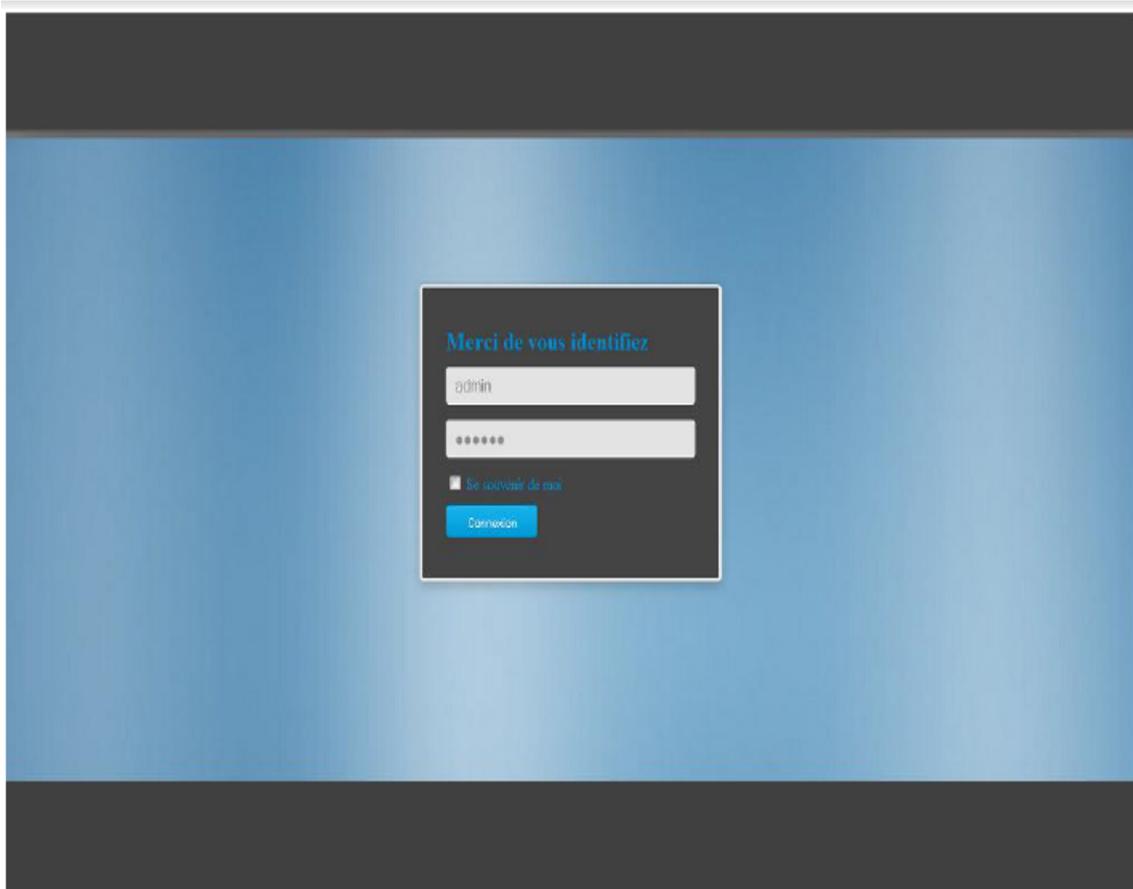


FIGURE 4.5 – Page authentication

L'utilisateur doit remplir les deux champs d'authentification en saisissant les coordonnées correctes afin qu'il puisse accéder à la page d'accueil.

4.4.2 Capture d'écran : Page d'accueil



FIGURE 4.6 – Page d'accueil

Après que l'utilisateur ait saisi ses informations, il peut maintenant accéder à la page d'accueil, ou il peut faire ce qu'il veut, il n'aura qu'à choisir dans le menu à gauche.

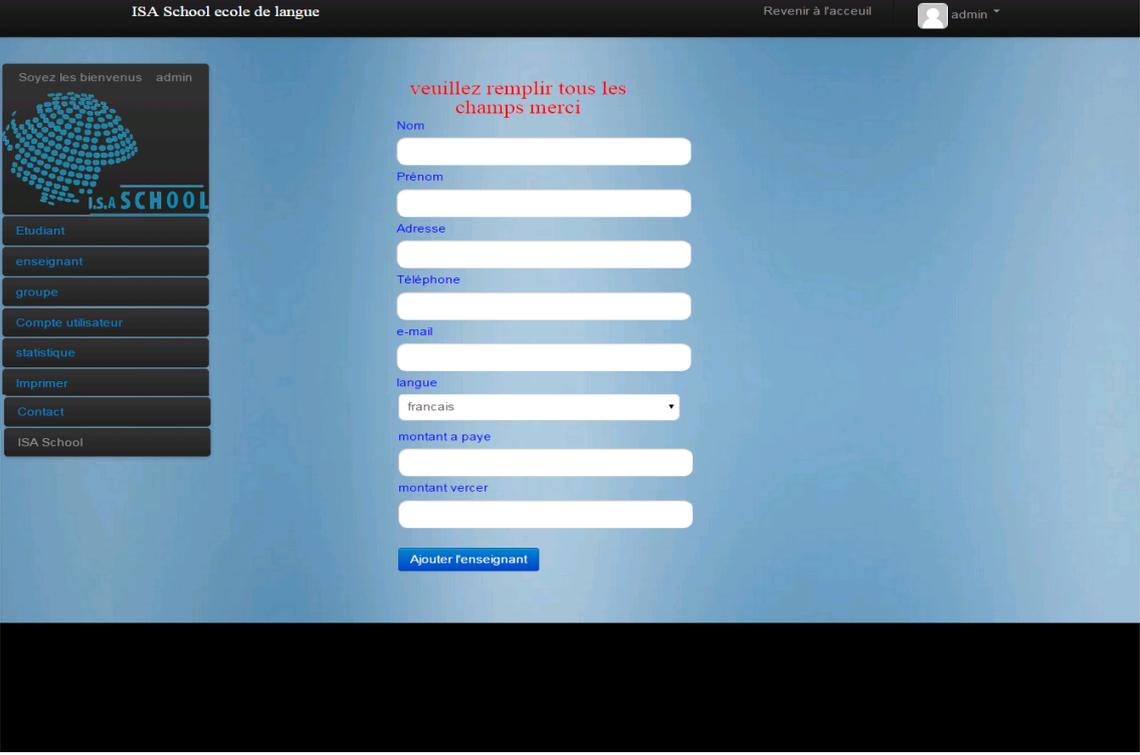
4.4.3 Capture d'écran : ajouter un dossier étudiant

The screenshot shows the 'ajouter un dossier étudiant' page in the ISA School system. The page has a dark blue header with the text 'ISA School ecole de langue' on the left and 'Revenir à l'accueil' and a user profile icon labeled 'admin' on the right. Below the header, there is a sidebar on the left with a dark background and white text, containing a welcome message 'Soyez les bienvenus admin' and a grid of navigation buttons: 'Etudiant', 'enseignant', 'groupe', 'Compte utilisateur', 'statistique', 'Imprimer', 'Contact', and 'ISA School'. The main content area has a light blue background and contains a form with the following fields: 'Nom' (text input), 'Prénom' (text input), 'date de naissance' (three dropdown menus for day, month, and year, with values 01, janvier, and 1960), 'lieu de naissance' (text input), 'Statut' (dropdown menu with 'Etudiant' selected), 'niveau scolaire' (text input), 'Adresse' (text input), 'Téléphone' (text input), 'note obtenue au test d'entrée' (dropdown menu with '1' selected), 'groupe' (dropdown menu with 'aagg /espagnol(Alphabétisation)' selected), 'montant a paye' (text input), and 'versement' (text input). At the bottom of the form is a blue button labeled 'Ajouter l'étudiant'.

FIGURE 4.7 – Page ajouter un dossier étudiant

Si l'utilisateur est dans ce cas, alors il doit remplir le formulaire d'ajout qui appartient à un étudiant. S'il manque des champs à remplir, un message d'erreur s'affiche sur l'écran.

4.4.4 Capture d'écran : ajouter un dossier enseignant avec champs vide



The screenshot shows the ISA School website interface. At the top, the header includes "ISA School ecole de langue" on the left, "Revenir à l'accueil" in the center, and a user profile icon labeled "admin" on the right. A navigation menu on the left side contains the following items: "Soyez les bienvenus admin", "Etudiant", "enseignant", "groupe", "Compte utilisateur", "statistique", "Imprimer", "Contact", and "ISA School". The main content area features a form titled "veuillez remplir tous les champs merci" in red text. The form fields are: "Nom", "Prénom", "Adresse", "Téléphone", "e-mail", "langue" (a dropdown menu currently set to "français"), "montant a paye", and "montant vercer". A blue button labeled "Ajouter l'enseignant" is positioned at the bottom of the form.

FIGURE 4.8 – Page ajouter un dossier enseignant

Si l'utilisateur est dans le cas d'ajouter un dossier enseignant, et ne remplit le formulaire d'ajout qui appartient à un enseignant, le message d'erreur "veuillez remplir tous les champs merci" s'affiche sur l'écran.

4.4.5 Capture d'écran : ajouter un groupe existant

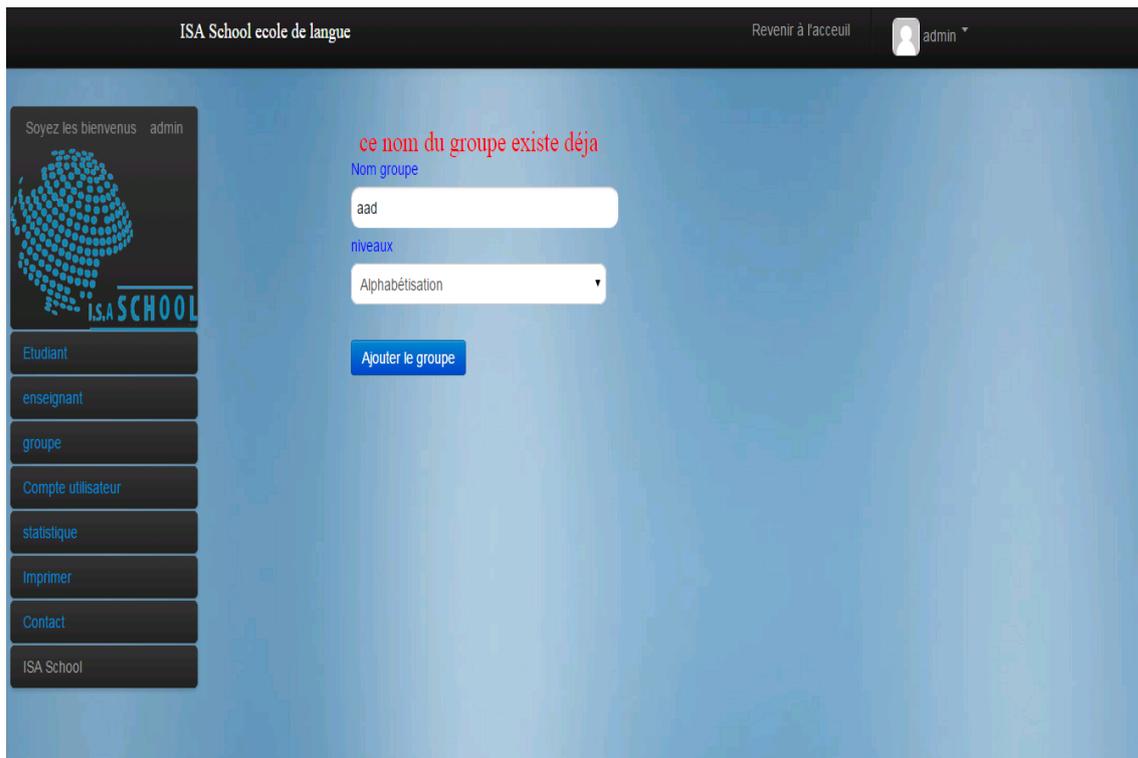


FIGURE 4.9 – Page ajouter un groupe

Si l'utilisateur est dans le cas d'ajouter un groupe, et que le groupe existe déjà alors le message d'erreur "ce nom du groupe existe déjà" s'affiche sur l'écran.

4.4.6 Capture d'écran : modifier un dossier étudiant

ISA School ecole de langue

Revenir à l'accueil admin

Soyez les bienvenus admin

Etudiant

enseignant

groupe

Compte utilisateur

statistique

Imprimer

Contact

ISA School

Nom
ouy

Prénom
abdelatif

Status
Etudiant

Adresse
tala merkha

Téléphone
0697046954

son groupe
aagg /espagnol(Alphabétisation)

nouveau groupe
aagg /espagnol(Alphabétisation)

montant a paye
6000

vercement

modifier l'etudiant

FIGURE 4.10 – Page modifier un dossier étudiant

Si l'utilisateur choisit de modifier un étudiant le système lui affiche le formulaire de l'étudiant rempli, il saisi les modifications nécessaires et confirme la modification.

4.4.7 Capture d'écran : supprimer un dossier étudiant

ISA School ecole de langue

Revenir à l'accueil admin

Soyez les bienvenus admin

identifiant
82

Nom
ouy

Prénom
abdelatif

supprimer l'etudiant

Etudiant

enseignant

groupe

Compte utilisateur

statistique

Imprimer

Contact

ISA School

FIGURE 4.11 – Page supprimer un dossier étudiant

Si l'utilisateur choisit de supprimer un étudiant le système lui affiche le formulaire de suppression, il saisi les informations nécessaires et supprime.

4.4.8 Capture d'écran : confirmer la suppression

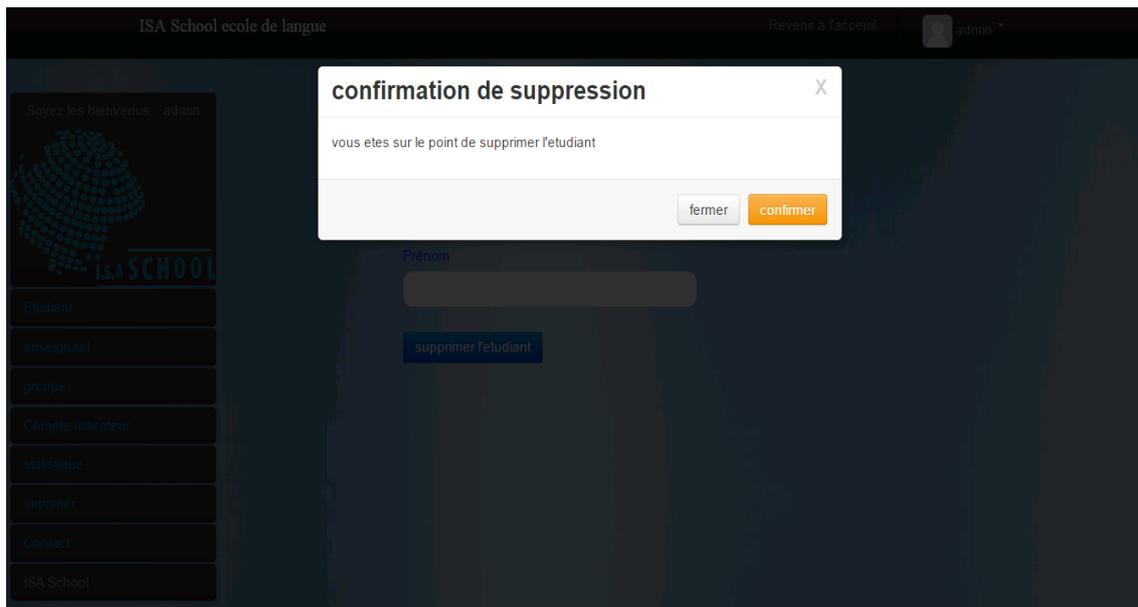


FIGURE 4.12 – Page confirmer la suppression

Pour plus de sécurité à la suppression d'un dossier, une petite fenêtre apparait à l'utilisateur pour qu'il confirme s'il est sur de vouloir supprimer.

4.5 Conclusion

Cette étape à mener à l'existence de ce qui était conceptuel. En effet nous avons décrit le prototype de la réalisation et nous avons mis l'accent sur la programmation de notre application web dédié à la gestion de l'école ISA School, et nous avons terminé ce chapitre par la présentation de quelques pages d'écran de l'application.

Conclusion générale et perspectives

Notre projet consiste à concevoir et mettre en œuvre une application web de gestion des étudiants de l'école des langues étrangères ISA School.

Dans ce travail, nous avons parlé de la gestion du centre de formation ISA School, et nous avons présenté les étapes de développement de notre application.

Nous nous sommes appuyé également durant tout le processus de conception sur l'UML, qui est un outil graphique pratique pour illustrer notre démarche.

Le but de l'application était de montrer l'intérêt de l'informatisation du secteur éducatif et scolaire, en réalisant un prototype de gestion des dossiers des étudiants d'ISA School, en assurant plusieurs avantages par rapport à la gestion manuelle à savoir :

- Faciliter l'exercice professionnel quotidien des différents professionnels d'ISA School.
- Améliorer le stockage, l'archivage et diminuer le temps d'accès aux dossiers.

L'application réalisée permet d'assurer plusieurs fonctionnalités de base à savoir la création d'un dossier, sa modification, consultation des différents dossiers, l'impression des différents documents et le calcul des statistiques.

Le Thème qui nous a été attribué est très instructif sur le plan pédagogique et très intéressant sur le plan technologique et développement. Nous en tant qu'étudiants en fin de cycle II nous a permis de :

- Accroître nos connaissances.
- Initier aux différentes technologies de développement (css, php...).
- Améliorer nos compétence dans la programmation orienté objet.

Au final étant donné que nul ne peut se prétendre aborder un domaine dans son ensemble nous souhaiterons venir :

- Améliorer les interfaces pour quelles répondent aux critères ergonomique.
- Etablir un système de sécurité des bases de données et limiter le nombre de tentatives d'authentification à l'application.
- Héberger l'application sur un serveur.
- Développer l'application à une application mobile.

Annexes a

Cahier de charge

Le cahier des charges se définit comme un acte, un document de référence qui permet à un dirigeant d'entreprise de préciser les conditions, les règles et les exigences d'une mission, d'une intervention, d'un travail à accomplir ou d'une tâche à exécuter par un consultant en management, en vue de résoudre un problème spécifique ou d'améliorer une situation donnée, tout en déterminant les résultats attendus.

Objectifs du cahier de charge

Le cahier des charges est une demande de service détaillé, élaborée de façon à protéger les intérêts du dirigeant d'entreprise et à améliorer la qualité de l'offre présentée par le consultant.

Il permet plus particulièrement :

- de définir précisément le problème ou la situation à améliorer ;
- d'identifier clairement les besoins de l'entreprise ;
- de préciser les objectifs de l'intervention et les résultats attendus ;
- d'établir les règles et les exigences de base pour la réalisation de l'intervention ;
- d'obliger la société-conseil et ses conseillers à préciser les services à fournir et les biens livrables ;
- de présenter les éléments d'une entente pouvant lier les deux parties ;
- de servir de document de référence pour éviter les problèmes d'interprétation et les outils ;
- de définir les responsabilités et les rôles respectifs des parties engagées dans l'intervention ;

Annexes b

Les diagramme de l'UML

Diagrammes structurels (statiques) :

1. Diagramme de classes : il représente les classes intervenant dans le système.
2. Diagramme d'objets : il sert à représenter les instances de classes (objets) utilisées dans le système.
3. Diagramme de composants : 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...).
4. Diagramme de déploiement : 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.
5. Diagramme des paquetages : 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.
6. Diagramme de structure composite : permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.

Diagrammes comportementaux(dynamique)

1. Diagramme des cas d'utilisation : il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
2. Diagramme états-transitions : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.
3. Diagramme d'activité : permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.
4. Diagramme de séquence : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
5. Diagramme de communication : représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
6. Diagramme global d'interaction : permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
7. Diagramme de temps : permet de décrire les variations d'une donnée au cours du temps.

Annexes c

La réalisation du système 2TUP consiste à fusionner les résultats des deux branches Une forme en Y

Du coté de la branche fonctionnelle

Capture des besoins fonctionnels : elle aboutit à un modèle des besoins focalisé sur le métier des utilisateurs. Elle minimise le risque de produire un système inadéquat avec les besoins des utilisateurs. De cette capture, la MOE consolide les spécifications et en vérifie la cohérence et l'exhaustivité.

Analyse : étude des spécifications afin de savoir ce que le système va réellement réaliser en termes de métier. Découpage en composants.

Du coté de la branche technique

Capture des besoins techniques : recensement des outils, des matériels et des technologies à utiliser; des contraintes (temps de réponse maximal, contraintes d'intégration avec l'existant) tout cela va aboutir à une première conception de l'architecture technique.

Conception générique : Découpage en composants nécessaires à la construction de l'architecture technique. Il est généralement conseillé de réaliser un prototype pour assurer la validité de l'architecture. Cette étape permet de minimiser l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles

la branche du milieu

Conception préliminaire : étape délicate durant laquelle on intègre le modèle d'analyse dans l'architecture technique. Le but ici est de savoir dans quel composant technique on met nos fonctionnalités issues de l'analyse.

Conception détaillée : conception de chaque fonctionnalité.

Etape de codage : phase de programmation de ces fonctionnalités, avec des tests au fur et à mesure.

Etape de recette : phase de validation des fonctions du système développé[22].

Bibliographie

- [1] <http://www.memoireonline.com/ApplicationwebdeGestiondepharmacieenJava2.html> Date de consultation 22/05/2015
- [2] <http://html5-tutorial.fr/developpementd'applicationweblapprochecomposant\etlapprochemodulaire/> Date de consultation 7/05/2015
- [3] <http://www.aidice--web.com/accueil/definitionsiteweb.php> Date de consultation 7/05/2015
- [4] <https://www.ideematic.com/dictionnaireweb/applicationweb> Date de consultation 7/05/2015
- [5] <http://www.Developpez.com/LelangagePHP> Date de consultation 28/06/2015
- [6] <http://www.Developpez.com/UP:unifiedprocessus> Date de consultation 22/05/2015
- [7] [https://msdn.microsoft.com/fr/library/aa292164\(\\$v=vs.71\\$\).aspx](https://msdn.microsoft.com/fr/library/aa292164($v=vs.71$).aspx) Date de consultation 7/05/2015
- [8] <http://aws.amazon.com/fr/application-hosting/> Date de consultation 13/06/2015
- [9] <http://forums.cnetfrance.fr/topic/125302-hebergement-application-web-php-repondu/> Date de consultation 13/06/2015
- [10] <http://www.microsoft.com> Date de consultation 03/04/2015
- [11] <http://www.wampserver.com/> Date de consultation 03/04/2015
- [12] <https://developer.mozilla.org/fr/docs/web--technologieweb\pourdeveloppeurs--css> Date de consultation 03/04/2015
- [13] <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/sequence-diagram> Date de consultation 03/06/2015
- [14] <http://www.linformaticien.com/dossiers/id/28420/pageid/19543/le-comparatif-2013-de-21-hebergeurs.aspx> Date de consultation 13/06/2015
- [15] Samir ADOUANE. Intégration des moyens de modification dynamique des contenus sur le web. Mémoire de magister, Université el Hadj Lakhdar Batna, Batna, 2007.
- [16] Philip BOUSQUET. Créer une application web avec PHP, 2008.
- [17] L. SHKLAR et R. ROSEN. Architecture : Principles, Protocols and Practices. Edition :John WILEY et SONS, 2003.
- [18] C. PORTENEUVE et T. NITOT. Bien développer pour le web 2.0 : Bonnes pratiques Ajax. Eyrolles, 2008.
- [19] Joseph GABAY. UML2 analyse et conception,Mise en oeuvre guidée avec études de cas. DUNOD, 2008.

- [20] Rahmouna SAMBA Soumia HADJ JELLOUL . Suivi des enseignants du lmd par l'application de la méthode 2tup.mémoire Master2, Université Béchar,2009.
- [21] J.CONALLEN. Concevoir des applications web avec UML. Eyrolles, 2000.
- [22] I.JOCOBSON G.BOOCH J.RUMBAUGH. Le processus unifié de développement(the unified software development process). EYROLLES, 2000.
- [23] David LACERTE. Applications web et mobiles. TAURON, 25septembre 2013.
- [24] Robert OGOR. Modélisation avec uml. [http ://www.enstbretagne.com](http://www.enstbretagne.com), mai 2003.
- [25] Pascal ROQUES Franck VALLEE,UML2 en action,Eyrolles 4 eme edition,2007.
- [26] Aurélien GERON Pierre-Yves CLOUX, David DOUSSOT. Technologies et architectures Internet Corba, COM, XML, J2EE, .NET, web services. DUNOD, Paris, 2 edition,mai 2003.
- [27] R.DAVID. A History Of The Dynamic Web. [http ://royal.pingdom.com](http://royal.pingdom.com), décembre 2007.
- [28] Philippe RIGAUX. Cours de bases de données. juin 2001.
- [29] Pascal ROQUES. Les Cahiers du Programmeur UML 2 Modéliser une application web. ISBN : 978-2-212-12389-0. 4eme édition,Groupe Eyrolles, 2008.

Résumé

Dans le monde contemporain, l'information est devenue le principal moteur de progrès économique et social. L'importance rôle qu'elle joue dans une organisation lui fait consacrer des investissements important et une attention particulière. Notre travail consiste à concevoir et réaliser une application web de gestion des étudiants d'une école privée pour les langues étrangères. Pour ce faire, nous avons suivi la démarche de développement logiciel UP et le langage de modélisation UML. La mise en œuvre de notre application s'articule sur une base de données développée avec Mysql selon une architecture trois tiers. L'environnement de développement utilisé est le wamp.

Mots-clés : processus UP, 2TUP, UML, PHP, Wamp, CSS3, HTML5, architecture 3 tiers.

Abstract

In today's world, information has become the main engine of economic and social progress. The important role it plays in an organization made him devote significant investment and attention. Our job is to design and develop a web application for managing student of a private school for foreign languages. To do this, we followed the UP software development approach and the UML modeling language. The implementation of our application is based on a database developed with MySQL using a three-tier architecture. The development environment is used PHP.

Keywords : UP process, UML, PHP, 3-tier architecture.