

**REPUBLIQUE DU SENEGAL**



**Un peuple-un but-une foi**

**Ministère de l'Enseignement Supérieur, de la Recherche et de l'innovation**

**Direction de l'Enseignement Supérieur Privée**

**Institut Supérieur d'Informatique**

**ISI**

**Mémoire de fin de cycle pour l'obtention de la licence professionnelle  
en Informatique Appliquée à la Gestion des Entreprises**

**(IAGE)**

**MISE EN PLACE D'UNE PLATEFORME DE PRE-  
INSCRIPTION UNIVERSITAIRE POUR DES NOUVEAUX  
BACHELIERS**

**Présenté et soutenu par :  
M. BAH El Hadj Ibrahima**

**Sous la direction de  
Mme DIAGNE AICHA  
Ingénieur en Génie Logiciel**

**Année Académique : 2021-2022**



## Dédicace

Je rends tout d'abord grâce à ALLAH le clément et le miséricordieux louange à ALLAH le tout puissant.

Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement :

➤ À mes chers parents

En guise de gratitude pour leur amour, leur patience, leur soutien, leurs encouragements et leurs sacrifices, aucun mot, aucune dédicace ne pourrait exprimer notre respect, notre considération ainsi que l'expression de notre profond amour. Puisse Dieu les accorder santé, bonheur et longue vie afin que nous puissions un jour combler de joie leurs vieux jours.

➤ À mes frères et sœurs

Quand j'ai besoin de courage et de force c'est vers eux que je me tourne toujours. C'est cette confiance aveugle qu'ils ont pour moi qui me motive, qui me pousse à toujours me surpasser et à dépasser mes limites.

- À tous les amis, que j'ai connus jusqu'à maintenant. Merci pour leur amour et leurs encouragements.
- À mes camarades de classe, en souvenir des années et les plus agréables moments que nous avons partagé. Pour toute la complicité et l'entente qui nous unissent, ce travail est un témoignage de mon attachement et de mes sentiments les meilleurs.
- À toutes les autres personnes aussi qui de près ou de loin ne cessent de me soutenir et de m'encourager.
- A tous mes professeurs qui m'ont permis d'apprendre durant tout ce cursus au près des quels j'ai acquis toutes ces compétences qui m'ont permis aujourd'hui de vous présenter ce travail.

## **Remerciements**

Je rends grâce à Allah le tout puissant qui nous a permis de mener à bout ce travail qui a été soumis à votre appréciation, ensuite nous adressons nos remerciements les plus chaleureux à :

- Notre encadreur Mme DIAGNE Aïcha, Ingénieur en génie logiciel, pour sa disponibilité et ses conseils qu'elle nous donne sans pour autant se lasser, merci pour tout Mme.
- L'ensemble du personnel du département Génie Informatique de l'ISI ayant participé d'une manière ou d'une autre à notre formation de licence.

Enfin nos remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année Universitaire...

## Avant-propos

L'Institut Supérieur d'informatique (ISI) est un institut d'enseignement supérieur avec une expérience de plus de dix ans dans la formation continue de type académique. ISI dispose d'un bureau qui s'occupe des études de conception, d'expérimentation, de réalisation et de conseil dans différents domaines. Le groupe ISI offre plusieurs formations et délivre les diplômes suivants : DTS, LICENCE, MASTER et éventuellement le DOCTORAT. Parmi les filières offertes par ISI, il y en a dans le domaine de l'informatique, des réseaux, de la gestion et l'organisation des entreprises.

L'obtention de la licence en informatique appliquée à la gestion d'entreprise, ISI exige aux étudiants la rédaction d'un mémoire de fin de cycle. C'est dans ce cadre que nous avons élaboré ce document qui a pour sujet : **mise en place d'une plateforme de pré-inscription universitaire pour des nouveaux bacheliers.**

En effet, cette plateforme a pour but spécifique de permettre aux futur(e)s étudiants d'avoir des choix divers et d'avoir une facilité dans l'obtention d'une université, de réduire les activités de recherches et d'octroyer aux acteurs concernés une sérénité sur leur orientation.

Ce document constitue notre premier travail de recherche académique, c'est pourquoi nous sollicitons de la part du jury, beaucoup d'indulgence pour ce qui concerne son évaluation.

# Sommaire

Dédicace.....	i
Remerciements.....	ii
Avant-propos .....	iii
Sommaire .....	iv
Glossaire .....	v
Liste des figures .....	vi
Liste des tableaux .....	vii
Résumé.....	viii
Abstract .....	ix
INTRODUCTION GENERALE .....	1
1. Le cadre théorique et méthodologique.....	3
1.1. Le cadre théorique .....	3
1.2. Le cadre méthodologique.....	4
2. Cadre organisationnel et conceptuel .....	13
2.1. Cadre organisationnel.....	13
2.2. Cadre conceptuel .....	14
3. Implémentation du logiciel.....	28
3.1. Présentation des types d'architectures .....	28
3.2. Réalisation de l'application .....	30
Conclusion générale .....	38
Reference bibliographique .....	39
Webographie.....	40
Table des matières .....	41

## **Glossaire**

**CLR** : Common Langage Runtime

**CSS** : Cascading Style Sheets(anglais), feuilles de style en cascade (français)

**HTML**: HyperText Markup Language

**OMT**: Object Modeling Technique

**OOSE**: Object Oriented Software Engineering

**PHP**: HyperText Preprocessor

**POO** : Programmation Orientée Objet

**SGBD** : Système de Gestion de Base de Données

**UML**: Unified Modeling Language

**XML**: Extensible Markup Language

## Liste des figures

Figure 1: : Cycle d'abstraction .....	6
Figure 2: Illustration diagramme de cas d'utilisation.....	10
Figure 3: Illustration diagramme de classe.....	11
Figure 4: Illustration diagramme de séquence .....	11
Figure 5: Diagramme de cas d'utilisation visiteur .....	15
Figure 6: Diagramme de cas d'utilisation étudiant .....	16
Figure 7: Diagramme de cas d'utilisation service gestion .....	16
Figure 8: Séquence d'authentification.....	20
Figure 9: Séquence créer ou modifier dossier .....	21
Figure 10: Séquence envoie de demande.....	22
Figure 11: Séquence validation de demande.....	23
Figure 12: Diagramme de class globale.....	25
Figure 13: Diagramme de déploiement .....	26
Figure 14: Architectures un-tiers.....	28
Figure 15: Architecture deux-tiers .....	29
Figure 16: Architecture trois-tiers.....	30
Figure 17: Interface administrateur .....	35
Figure 18: Interface d'accueil.....	35
Figure 19: Interface des filières.....	36
Figure 20: Consultation d'une filière.....	36
Figure 21: Interface renseignement.....	37



## Liste des tableaux

<b>Tableau 1: Modèle d'analyse.....</b>	<b>7</b>
<b>Tableau 2: Acteurs et cas d'utilisations.....</b>	<b>14</b>
<b>Tableau 3: Description textuelle valider demande.....</b>	<b>17</b>
<b>Tableau 4: Description textuelle créer dossier .....</b>	<b>18</b>
<b>Tableau 5: Description textuelle gérer école.....</b>	<b>19</b>

## Résumé

Si la mondialisation est un phénomène assez ancien, les tournures qu'elle prend aujourd'hui sont inédites. Elle touche tous les domaines y compris ceux qui ont toujours été hors du champ de la commercialisation. Cette globalisation s'est traduite par une libéralisation progressive et intense à laquelle n'échappe pas l'enseignement en général, l'enseignement supérieur en particulier.

Les principes et méthodes du monde des affaires gagnent de plus en plus le milieu éducatif. Au Sénégal, l'enseignement supérieur connaît un développement fulgurant. Les filières et les diplômes grouillent alors que la qualité des prestations des établissements n'est pas souvent évaluée.

Ce travail consiste à concevoir et à mettre en place une plateforme web d'inscription pour les nouveaux bacheliers afin de faciliter les modalités de recherches et d'inscription et permettre aux futur(e)s étudiants d'avoir des choix divers et d'avoir une aisance dans l'obtention d'une université.

## **Abstract**

If globalization is a fairly old phenomenon, the turns it is taking today are unprecedented. It affects all areas including those that have always been outside the scope of marketing. This globalization has resulted in a progressive and intense liberalization which does not escape education in general, higher education in particular.

The principles and methods of the business world are increasingly gaining ground in education. In Senegal, higher education is experiencing rapid development. Courses and diplomas are teeming while the quality of the services of the establishments is not often evaluated.

This work consists of designing and setting up a web registration platform for new graduates. To facilitate research and registration procedures and allow future students to have various choices and to have ease in obtaining a university.

# INTRODUCTION GENERALE

L'éducation est fondamentale dans la société d'aujourd'hui. Elle est bien sûr d'abord l'œuvre de la famille dans laquelle l'enfant apparaît et du milieu dans lequel il fait ses premiers pas. Elle est enrichie par les rencontres, les engagements, les expériences, qu'il aura tout au long de son existence. Mais la collectivité ne peut s'en désintéresser. Elle doit en organiser le cadre, en fournir les moyens, veiller à ses résultats. Sur le chemin menant à la personnalisation de l'apprentissage, la technologie autonomise les élèves en leur permettant de s'impliquer. Elle offre une éducation adaptée à leur vie numérique et prépare leur avenir.

De nos jours, dans le secteur de l'éducation pré universitaire, l'orientation des nouveaux bacheliers devient une lourde tâche avec des moyens archaïques, des options de plus en plus nombreuses et des établissements qui ne finissent pas de naître. Comment mettre en place une plateforme d'inscription pré universitaire commune ? Qu'est ce qui pourrait faciliter les modalités de recherches et d'inscription ? Quels intérêts cette plateforme pourrait-elle apporter aux différents acteurs ?

De ce fait, le travail consiste à concevoir et à mettre en place une plateforme web d'inscription pour les nouveaux bacheliers. Ce site web a pour but spécifique de permettre aux futur(e)s étudiants d'avoir des choix divers et d'avoir une facilité dans l'obtention d'une université, de réduire les activités de recherches et d'octroyer aux acteurs concernés une sérénité sur leur orientation.

Le non-respect des modalités d'inscriptions, les énormes efforts fournis pour des recherches, l'influence sur le choix de filières sont des difficultés que l'on croise souvent dans la société. C'est pour pallier à cela que nous suggérons de mettre en place cette plateforme.

La recherche documentaire et l'enquête au niveau des établissements et différents sites constitueront notre moyen d'investigation pour la mise en œuvre de ce mémoire.

Ce présent mémoire sera constitué par trois (3) parties : tout d'abord la première parlera des cadres théoriques et méthodologiques, ensuite la deuxième constituera l'analyse et la conceptualisation du système pour enfin passer à la troisième et la dernière partie qui permettra de mettre en œuvre de la plateforme.

## **CADRE THEORIQUE ET METHODOLOGIQUE**

## **1. Le cadre théorique et méthodologique**

Dans cette première partie intitulée CADRE THEORIQUE ET METHODOLOGIQUE il s'agit de faire les points des travaux ayant fait l'objet d'étude sur l'entreprise et ensuite définir la méthodologie de l'étude. Cette partie comporte deux chapitres : cadre théorique et cadre méthodologique.

### **1.1. Le cadre théorique**

Dans ce chapitre nous allons présenter la plateforme AMNA ECOLE. Ainsi cette partie comportera quatre sections : présentation du cadre de travail, problématique, objectifs et intérêt du sujet et hypothèses .

#### **1.1.1. Présentation de la structure**

AMNA ECOLE est une plateforme web de pré-inscription en ligne, qui permet aux nouveaux bacheliers de rechercher la filière et les établissements susceptibles de leur fournir la bonne formation et d'envoyer des demandes d'inscriptions aux instituts concernés.

#### **1.1.2. Présentation du projet**

##### **1.1.2.1. Contexte**

L'enseignement supérieur au Sénégal est marqué par un accroissement considérable des effectifs d'étudiants au niveau du pays et des flux internationaux d'étudiants. Les grandes universités sont unanimement engagées dans la compétition pour capter ces flux.

L'idée d'une plateforme web disposant d'une vision d'ensemble de tous les établissements privés et/ou publics d'enseignement pré-universitaire en ligne afin de rendre encore plus facile les recherches des différents établissements universitaires et filières est une initiative louable.

##### **1.1.2.2. Problématique**

De nos jours, dans le secteur de l'éducation pré universitaire, l'orientation des nouveaux bacheliers devient une lourde tâche avec des moyens archaïques, des options de plus en plus nombreuses et des établissements qui ne finissent pas de naître.

Ainsi, compte tenu de la situation citée ci-haut, nous sommes parvenus à nous poser les questions suivantes :

- Comment mettre en place une plateforme d'inscription pré universitaire commune ?
- Qu'est ce qui pourrait faciliter les modalités de recherches et d'inscription ?
- Quels intérêts cette plateforme pourrait-elle apporter aux différents acteurs ?

### **1.1.2.3. Objectif**

Le but spécifique consiste à concevoir et réaliser une plateforme web d'inscription pour les nouveaux bacheliers.

La réalisation d'une telle plateforme nous permet d'atteindre des objectifs spécifiques suivants :

- Disposer d'une vision d'ensemble de tous les établissements privées et/ou publics d'enseignement pré-universitaire ;
- Suivre le statut de son dossier : l'application devrait permettre au bachelier d'avoir un suivi sur son dossier jusqu'à l'obtention de l'admission ;
- Avoir une facilité dans l'obtention d'une université ;
- Réduire les activités de recherches et octroyer aux acteurs concernés une sérénité sur leur orientation ;
- Doter le nouveau système de plusieurs critères de recherches afin que l'utilisateur puisse choisir celle qui convient au mieux à ses besoins ;
- Proposer une plateforme assez dynamique et adaptable pour pouvoir satisfaire les besoins futurs ;
- Faciliter les tâches des insertions pour remplir les données nécessaires, comme par exemple, des informations complémentaires nécessaires relatives au profil de l'utilisateur.

## **1.2. Le cadre méthodologique**

Dans ce chapitre nous allons présenter la méthodologie adoptée pour la mise en place de l'application AMNA ECOLE. Ainsi cette partie comportera trois sections qui seront : Approche fonctionnelle, Approche orientée objet et choix approche.

### **1.2.1. Etude de l'existant**

Ce travail ne saurait débiter sans avoir une analyse juste et précise sur l'existant. La tâche a été de rencontrer des différents acteurs du secteur de l'enseignements supérieurs notamment des gérant d'établissement, des étudiants qui sont plus ou moins liés au secteur. Après quoi, il nous a été rapporté les difficultés rencontrées dans la recherche d'une filière et d'en connaître les débouchés et les moyens archaïques utilisés dans la publicité des établissements.

### **1.2.1.1. Critique de l'existant**

La gestion publicitaire des établissements se fait de façon basique et couteux en utilisant des panneaux publicitaires, des flyers distribués au niveau des institutions scolaires ce qui constitue des dépenses financières ; également au niveau des recherches beaucoup d'efforts sont fournis parfois sans entière satisfaction. Certains bacheliers trouvent des filières sans pour autant connaître les débouchés liés à cette formation. Remédier à cela en utilisant une plateforme plus enrichie en informations qui disposera d'une multitude de choix concernant les instituts privés et/ou publics universitaires permettrait de réduire déjà les coûts sur la publicité, les efforts de recherches et d'en savoir encore plus sur les différents choix à faire.

### **1.2.1.2. Technique d'investigation**

Pour la collecte d'informations, deux techniques principales ont été utilisées à savoir : la recherche documentaire dans une large mesure et l'envoi des formulaires d'enquête via des sites.

### **1.2.1.3. Méthode d'analyse et choix de la méthode**

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie. Il est ainsi nécessaire de définir la notion de méthode.

Une méthode, dans le contexte informatique, peut être définie comme une démarche fournissant une méthodologie et des notations standards qui aident à concevoir des logiciels de qualité. Une méthode d'analyse assure donc la pertinence, la représentativité et la hiérarchisation des besoins.

L'analyse se focalise sur quatre grands types d'information :

A qui est destinée l'application ? Quelles sont les caractéristiques des utilisateurs finaux ? Quels objectifs professionnels cette application va apporter aux utilisateurs ? Dans quel contexte l'utilisation de cette application va-t-elle s'inscrire ? De quelle façon, ces utilisateurs parviennent-ils à réaliser ses objectifs ? Dans cette partie, nous allons présenter deux outils de modélisation à savoir MERISE et UML.

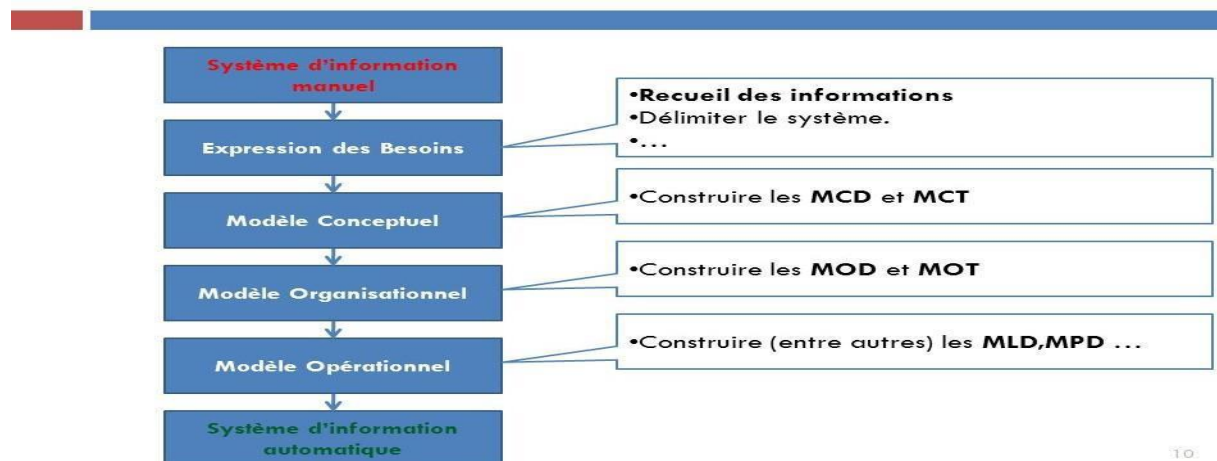
## **1.2.2. MERISE**

Merise propose une méthode de conception et de développement de Systèmes d'informations complète, détaillée, en grande partie formalisée, qui garantit (en principe) une



informatisation réussie. Elle fut créée en 1978 sous l'impulsion du Ministère français de la Recherche et de l'industrie. Dès 1980, c'est devenu un standard dans le domaine des systèmes intégrés de gestion. Ce système existe seulement dans les pays francophones. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. C'est d'ailleurs son point fort.

La conception du système d'information avec la méthode MERISE se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et traitements afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y ait pas de données superflues. Cette succession d'étapes est appelée cycle d'abstraction pour la conception des systèmes d'information :



*Figure 1 : Cycle d'abstraction*

Ainsi, la méthode MERISE s'appuie sur un certain nombre d'outils qui modélisent le système selon les niveaux d'analyse (abstraction) ou degré d'invariance :

**LE NIVEAU CONCEPTUEL** : s'attache aux invariants de l'entreprise ou de l'organisme du point de vue métier. Le compte rendu de ce niveau est de matérialiser, sous forme de dessins normalisés, de modèles complets par un dossier explicatif.

**LE NIVEAU LOGIQUE** : Ce niveau décrit la nature des ressources qui sont utilisées pour supporter la description statique et dynamique du système d'information. Ces ressources peuvent être humaines et/ou matérielles et logicielles. Il s'attache à préciser comment on organise les données de l'entreprise ou d'une organisation.

**LE NIVEAU PHYSIQUE** : permet d'établir la manière concrète dont le système sera mis en place. Il représente le résultat informatique, il dépend des logiciels de développement nécessaires à la programmation et la manipulation des données.

Niveau d'analyse	Données	Traitements
Niveau conceptuel	Modèle Conceptuel de Données (MCD)	Modèle Conceptuel de Traitements (MCT)
Niveau logique	Modèle Logique de Données (MLD)	Modèle Organisationnel de Traitements (MOT)
Niveau physique	Modèle Physique de données (MPD)	Modèle Physique de Traitements (MPT)

*Tableau 1: Modèle d'analyse*

**Modèle Conceptuel de Données (MCD)** : décrit la signification des données sur lesquelles reposent le système d'information et les structures.

**Modèle conceptuel de traitements (MCT)** : formalise les activités du domaine étudié.

**Modèle Logique de Données (MLD)** : fournit une description des données tenant compte des moyens informatiques mis en œuvre.

**Modèle Organisationnel de Traitements (MOT)** : décrit le fonctionnement du domaine étudié en présentant les ressources mises en œuvre et leur organisation.

**Modèle Physique de Données (MPD)** : est une description de la base de données ou de l'ensemble des fichiers correspondants aux données gérées par le système d'information. **Modèle physique de traitements (MPT)** : décrit les spécificités des différents modules de traitement.

### ➤ Avantages

Pour de petites bases de données, MERISE est généralement l'une des meilleures solutions du point de vue de l'architecture. Un des avantages de MERISE est quasiment de pouvoir être utilisé par un non informaticien. Ce qui permet de faire évoluer en temps réel la structure informatique d'une entreprise à mesure que celle-ci évolue.

MERISE est une méthodologie qui dispose de beaucoup d'outils de développement informatique tel que AMC designer, Power Designer, et est adaptée aux nouvelles technologies :

architecture client/serveur, interfaces graphiques, intranet/internet, démarche de développement rapide.

### ➤ Inconvénients

MERISE est un peu adapté pour les environnements distribués où plusieurs applications sont externes à un domaine d'interagir avec le modèle d'application. En outre, elle n'est pas en mesure de modéliser les données à caractère sémantique.

### 1.2.3. UML

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. 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 2.5.1.

UML définit 6 modèles pour la représentation des systèmes :

Modèle des classes : capture la structure statique,

Modèle des états : exprime le comportement dynamique des objets,

Modèle des cas d'utilisation : décrit les besoins de l'utilisateur,

Modèle d'interaction : représente les scénarios et les flots de messages,

Modèle de réalisation : montre les unités de travail,

Modèle de déploiement : précise la répartition des processus.

UML définit 9 diagrammes pour élaborer les 6 modèles :

Diagramme de classe : représentation de la structure statique en termes de classes et relations,

Diagramme de séquence : représentation temporelle des objets et leurs interactions,

Diagramme de collaboration : représentation spatiale des objets, des liens et des interactions,

Diagramme d'objet : représentation des objets et leurs relations (diagramme de collaboration simplifiée sans envois des messages),

Diagramme d'états-transitions : représentation du comportement d'une classe d'objet en termes d'états,

Diagramme d'activités : représentation d'une opération en termes d'actions,

Diagramme de cas d'utilisation : représentation des fonctions du système du point de vue de l'utilisateur,

Diagramme de composant : représentation des composants physiques d'une application,

Diagramme de déploiement : représentation du déploiement des composants sur les dispositifs matériels.

### ➤ **Avantage**

Étant donné qu'il est le fruit de la fusion de plusieurs méthodes objets, il utilise l'approche objet en présentant un langage de description universel. Il permet grâce à un ensemble de diagrammes très explicites, de représenter l'architecture et le fonctionnement des systèmes informatiques complexes en tenant compte des relations entre les concepts utilisés et l'implémentation qui en découle.

UML est un langage formel et normalisé, il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité UML est un support de communication performant car il cadre l'analyse et facilite la compréhension de représentations abstraites complexes Son caractère polyvalent et sa souplesse en font un langage universel.

### ➤ **Inconvénients**

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation. Un inconvénient que certains développeurs pourraient trouver en utilisant UML est le temps qu'il faut pour gérer et maintenir des diagrammes UML.

UML a une notation majoritairement graphique pouvant se révéler insuffisante ou trop chargée d'un point de vue expressivité et les liens sont parfois difficiles entre les vues et les diagrammes d'une même application.

## **1.2.4. Choix de la méthode**

La méthode Merise et le langage UML sont deux approches de modélisation d'un système d'information. Cela dit, la méthode Merise, s'arrête au niveau organisationnel pour les traitements et les données et ne s'occupe pas de l'interface utilisateur. Elle est très adaptée à un contexte de

création d'application mais pas forcément à un problème de maintenance ou de seconde informatisation.

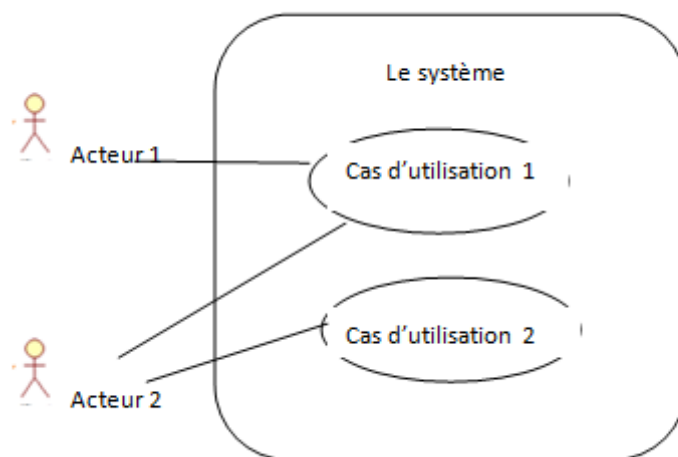
UML, contrairement à son prédécesseur MERISE qui pourtant est utilisée de nos jours, donne un sens intéressant à l'approche objet et couvre de plus tout le cycle de réalisation du logiciel. UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet.

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions. L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions. Son indépendance par rapport aux langages de programmation et aux domaines d'application ainsi aux processus, en fait un langage universel. UML est donc bien plus qu'un simple outil qui permet de "dessiner" des représentations graphiques... Il permet également de parler un langage commun, normalisé mais accessible.

C'est pourquoi UML a été choisi comme outil de modélisation de notre système.

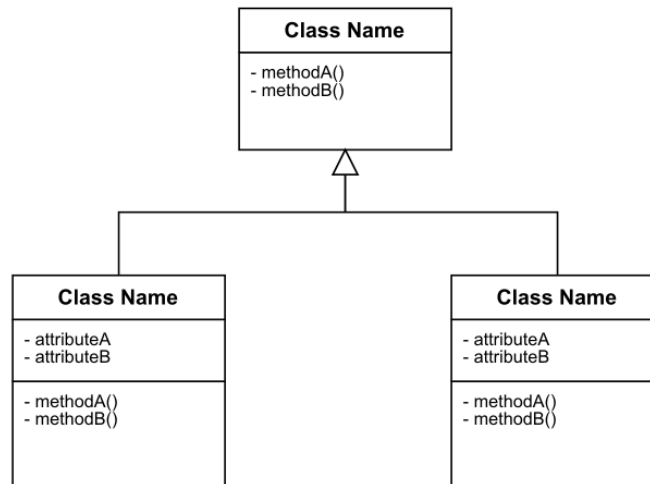
#### 1.2.4.1. Illustration de quelques diagrammes

- **Diagramme de cas d'utilisation**



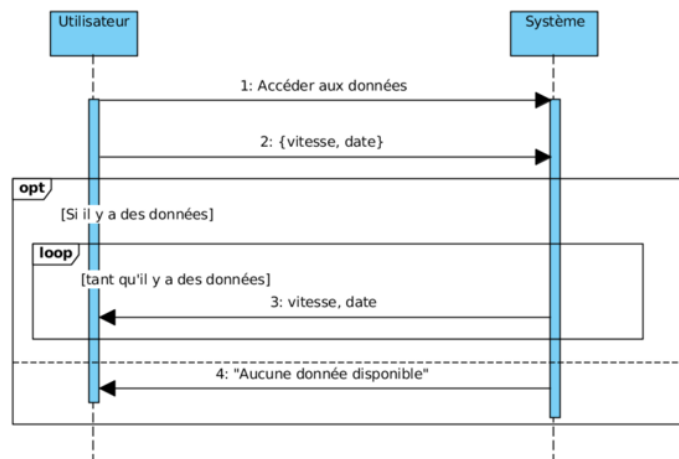
*Figure 2: Illustration diagramme de cas d'utilisation*

- Diagramme de classe



*Figure 3: Illustration diagramme de classe*

- Diagramme de séquence



*Figure 4: Illustration diagramme de séquence*

## **CADRE ORGANISATIONNEL ET CONCEPTUEL**

## 2. Cadre organisationnel et conceptuel

### 2.1. Cadre organisationnel

#### 2.1.1. Identification des besoins

L'ascension de tout projet d'étude repose sur la qualité de la phase de démarrage. Ainsi, l'analyse des besoins est une des premières techniques de gestion de projet à mettre en œuvre, d'autant plus qu'elle vise à identifier les exigences du projet et à faire le point sur les éléments attendus. Pour assurer ces objectifs, il est essentiel que nous parvenions à une vue claire des différents besoins escomptés de notre projet. Il faut déterminer au moindre détail les fonctionnalités attendues.

##### 2.1.1.1. Les besoins fonctionnels

Nous allons définir dans cette partie les besoins substantielles auxquelles notre application devrait répondre :

- Permettre à l'étudiant de créer un compte ainsi que son dossier ;
- Effectue des recherches de filières et d'établissements par l'étudiant ;
- Effectuer des demandes de préinscription par l'étudiant ;
- L'envoi de mail de confirmation par les gérants d'écoles ;
- La gestion des établissements et filières par l'administrateur ;

##### 2.1.1.2. Les besoins non-fonctionnels

Ils spécifient des critères qui jugent le fonctionnement d'un système, plutôt que des comportements spécifiques, par exemple : « les données modifiées dans une base de données doivent être mises à jour pour tous les utilisateurs qui y accèdent dans les 2 secondes ». Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

**Fiabilité :** L'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.

**Les ambiguïtés :** Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre plateforme.

**Ergonomie et bonne Interface :** L'application doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile) du point de vue navigation entre les différentes pages, couleurs et mise en textes utilisées.



**Sécurité :** Notre solution doit respecter surtout la confidentialité des données personnelles des Utilisateurs qui restent l'une des contraintes les plus importantes dans les plateformes de gestion immobilière.

**Aptitude à la maintenance et la réutilisation :** Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

### 2.1.2. Identifications des acteurs

Acteurs	Cas d'utilisations
Administrateur	<ul style="list-style-type: none"><li>• Gérer école</li><li>• Gérer filière</li></ul>
Gérant d'école	<ul style="list-style-type: none"><li>• Lister demande</li><li>• Valider demande</li></ul>
Visiteur	<ul style="list-style-type: none"><li>• Voir offre</li><li>• Créer compte</li></ul>
Etudiant	<ul style="list-style-type: none"><li>• Envoyer demande</li><li>• Gérer compte</li><li>• Voir offre</li></ul>

*Tableau 2: Acteurs et cas d'utilisations*

## 2.2. Cadre conceptuel

### 2.2.1. Point de vue fonctionnel

#### 2.2.1.1. Diagramme de cas d'utilisation

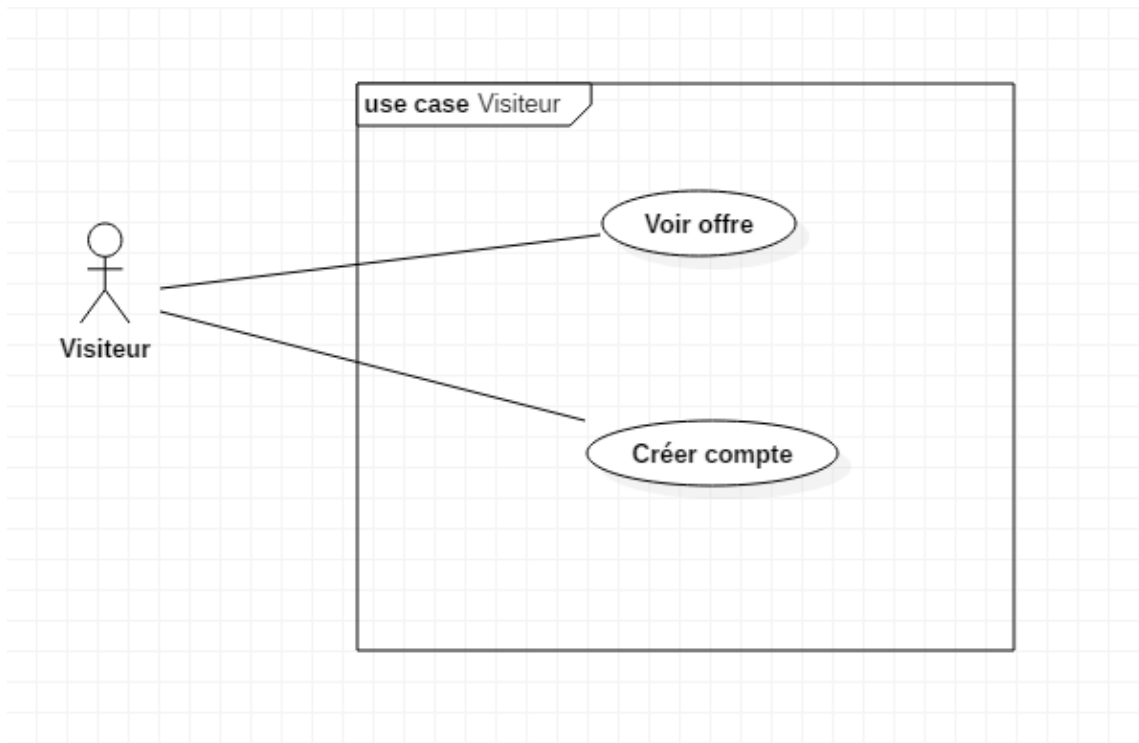
Le diagramme des cas d'utilisation (*Use Case Diagram*) constitue la première étape de l'analyse UML en :

- Modélisant les besoins des utilisateurs.
- Identifiant les grandes fonctionnalités et les limites du système.
- Représentant les interactions entre le système et ses utilisateurs.

Le diagramme des cas d'utilisation apporte une vision utilisateur et absolument pas une vision informatique. Il ne nécessite aucune connaissance informatique et l'idéal serait qu'il soit réalisé par le client.

Le diagramme des cas d'utilisations n'est pas un inventaire exhaustif de toutes les fonctions du système. Il ne liste que des fonctions générales essentielles et principales sans rentrer dans les détails.

➤ **Cas d'utilisation visiteur**



*Figure 5: Diagramme de cas d'utilisation visiteur*

## ➤ Cas d'utilisation étudiant

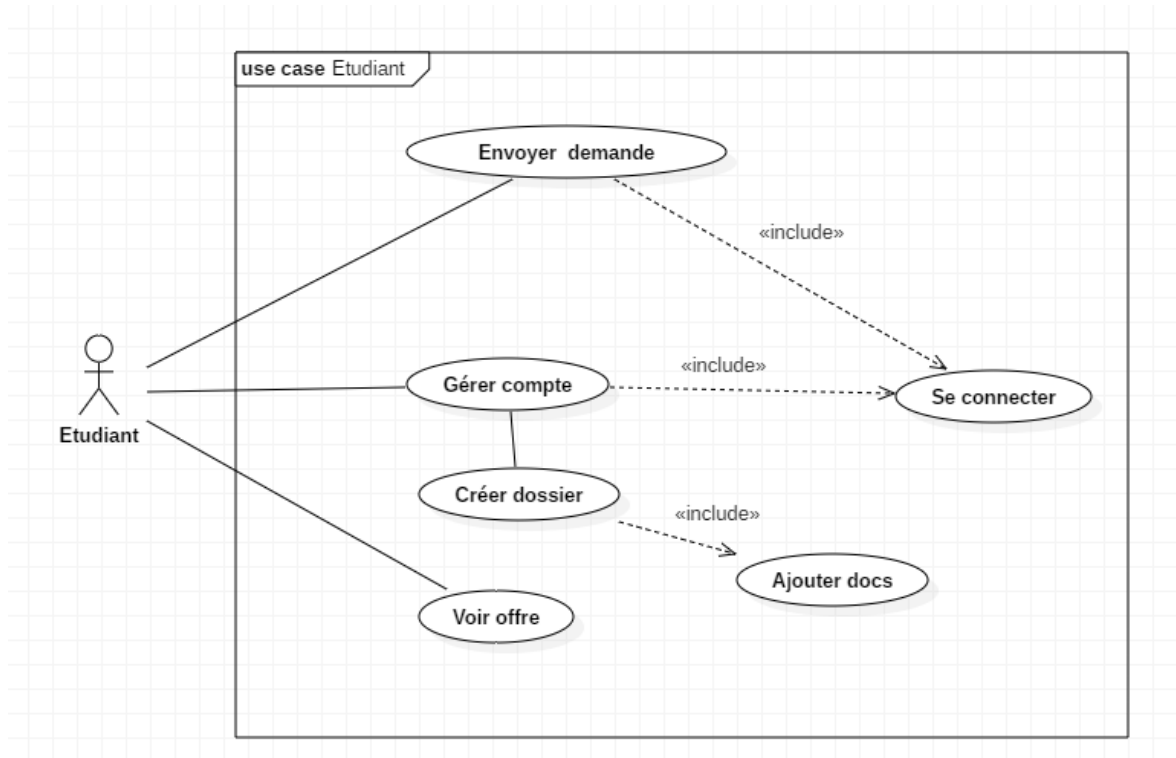


Figure 6: Diagramme de cas d'utilisation étudiant

## ➤ Cas d'utilisation service gestion

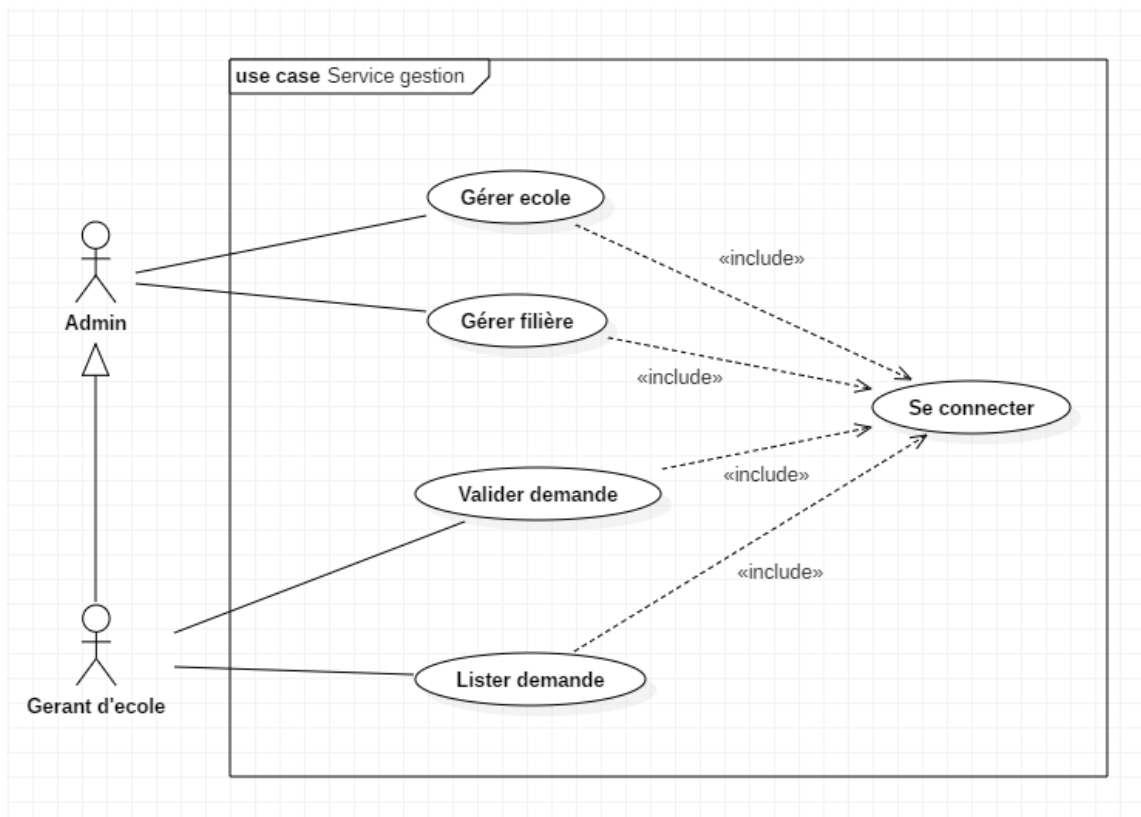


Figure 7: Diagramme de cas d'utilisation service gestion

### 2.2.1.2. Description Textuelle de quelques cas d'utilisation

<b>Cas d'utilisation</b>	Valider demande
<b>Acteur</b>	Gérant d'école
<b>Description</b>	Ce cas d'utilisation permet au gérant d'école de valider une ou des demandes
<b>Précondition</b>	L'utilisateur se connecte La liste des demandes est disponible
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1) Le système affiche l'interface de l'utilisateur selon son rôle ;</li><li>2) Le système retourne la liste des demandes de l'établissement concerné ;</li><li>3) L'utilisateur choisit une demande</li><li>4) L'utilisateur envoie une réponse de validation au concerné ;</li><li>5) Le système transmet la validation de la demande au concerné par e-mail.</li></ol>
<b>Postcondition</b>	Un mail de validation a été envoyé

Tableau 3: Description textuelle valider demande

<b>Cas d'utilisation</b>	Créer dossier
<b>Acteur</b>	Etudiant
<b>Description</b>	Ce cas d'utilisation permet à l'étudiant de créer son dossier
<b>Précondition</b>	L'utilisateur se connecte
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1) Le système affiche l'interface de l'utilisateur selon son rôle ;</li> <li>2) L'utilisateur choisit d'ajouter son dossier ;</li> <li>3) Le système affiche l'interface d'ajout d'un dossier ;</li> <li>4) L'utilisateur renseigne les informations nécessaires pour l'ajout d'un dossier et valide ;</li> <li>5) Le système transfère les nouvelles données dans la base et enregistre le dossier ;</li> </ol>
<b>Postcondition</b>	Le dossier a été ajouté

Tableau 4: Description textuelle créer dossier

<b>Cas d'utilisation</b>	Gérer école
<b>Acteur</b>	Administrateur
<b>Description</b>	Ce cas d'utilisation permet à l'administrateur de gérer les écoles
<b>Précondition</b>	L'utilisateur se connecte
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1) Le système affiche l'interface de l'utilisateur selon son rôle ;</li> <li>2) L'utilisateur choisit d'ajouter une école ;</li> <li>3) Le système affiche l'interface d'ajout d'une école ;</li> <li>4) L'utilisateur renseigne les informations nécessaires pour l'ajout d'une école et valide ;</li> <li>5) Le système transfère les nouvelles données dans la base et enregistre la nouvelle école ;</li> </ol>
<b>Postcondition</b>	Une école a été ajoutée

*Tableau 5: Description textuelle gérer école*

### 2.2.1.3. Diagramme de séquence

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

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

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

#### ➤ Séquence authentification

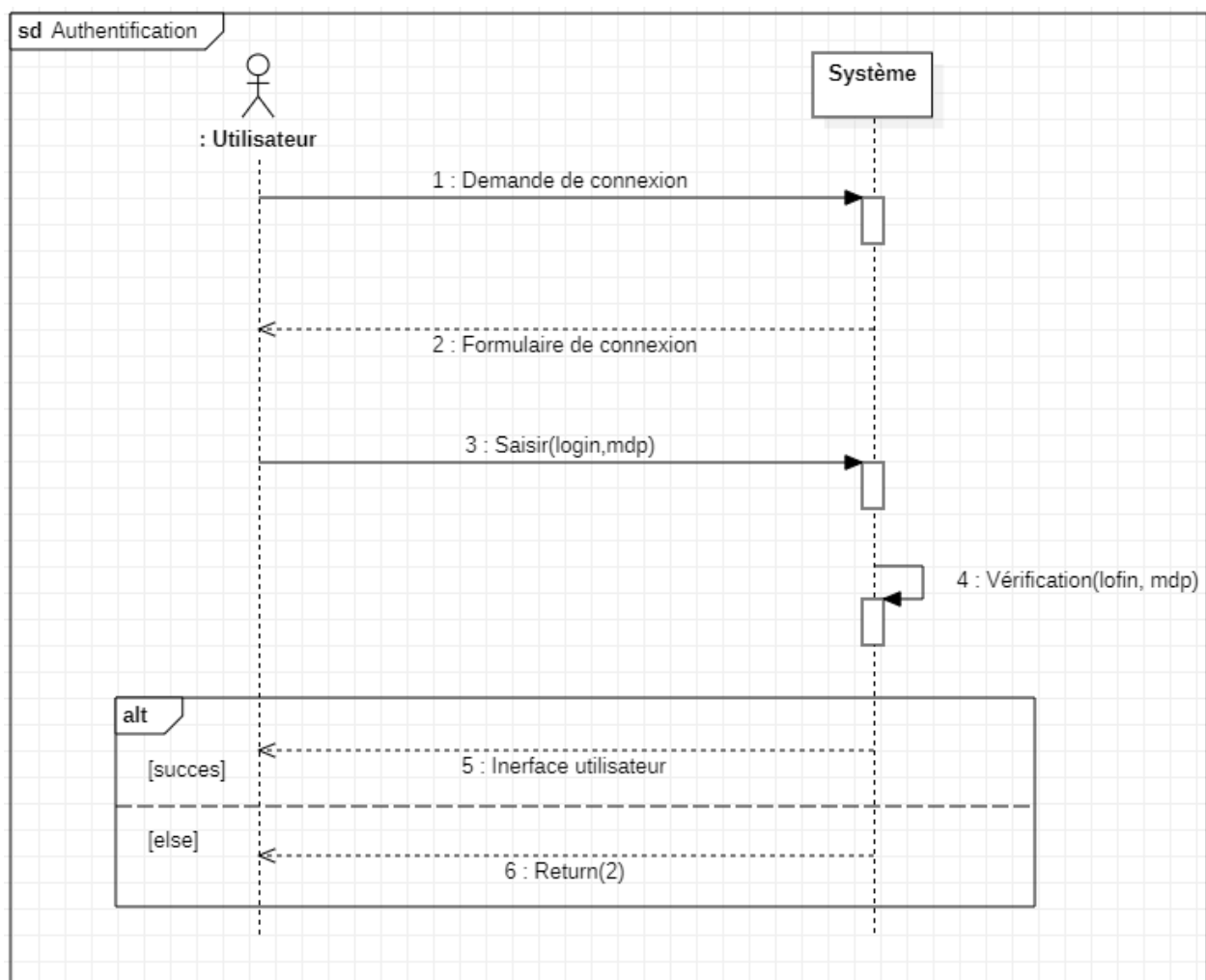


Figure 8: Séquence d'authentification

## ➤ Séquence création de dossier

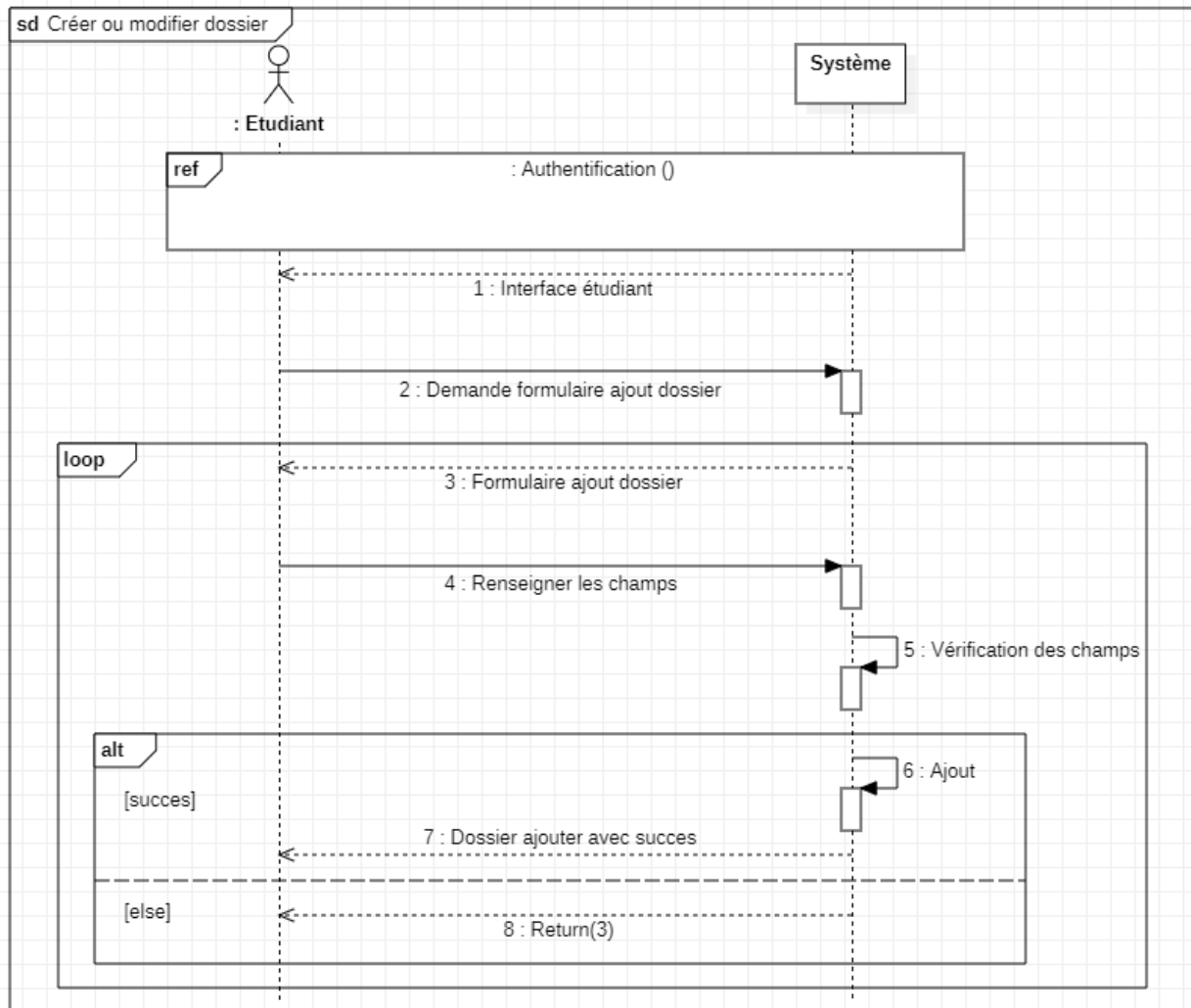


Figure 9: Séquence créer ou modifier dossier



## ➤ Séquence envoie de demande

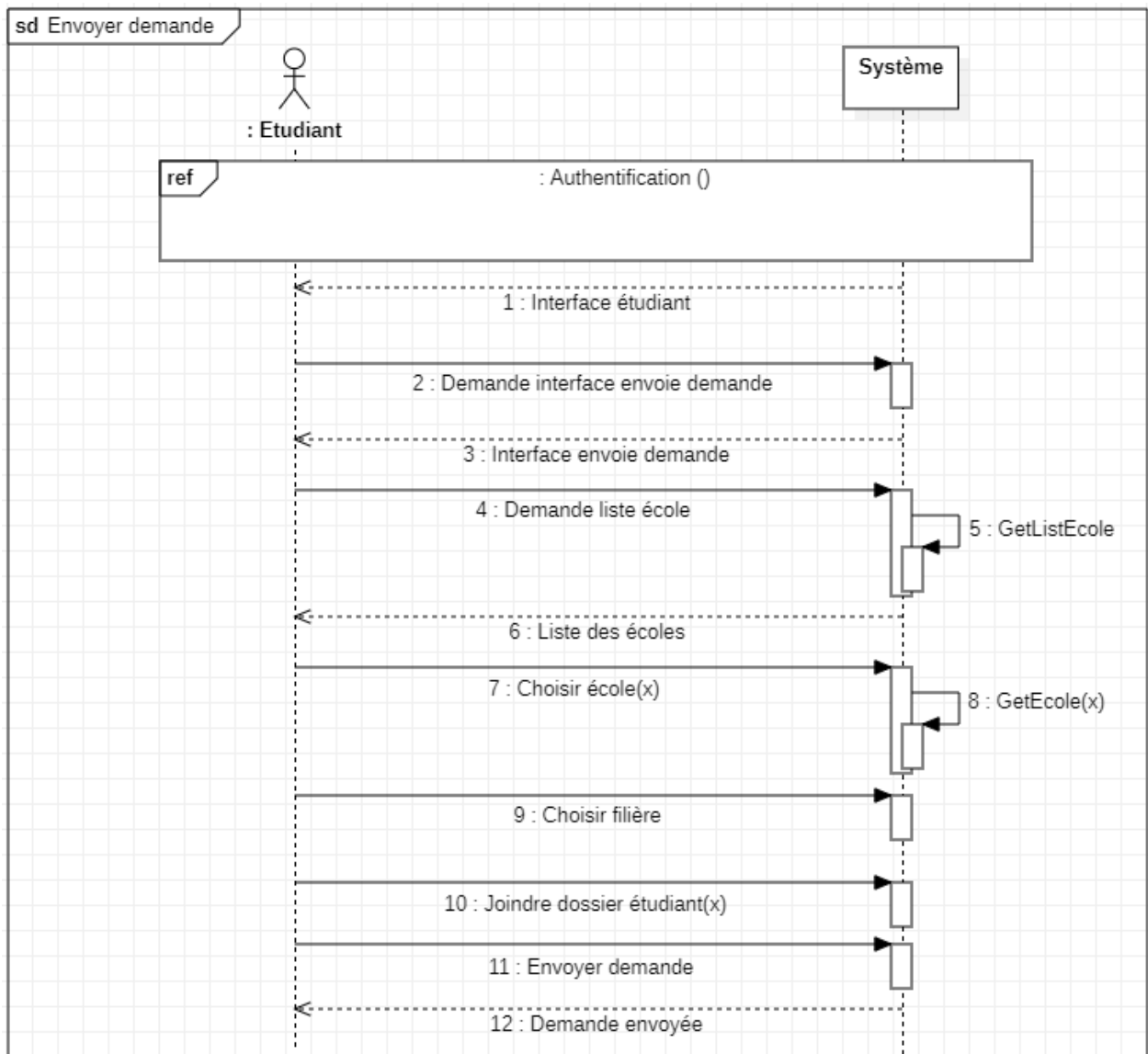
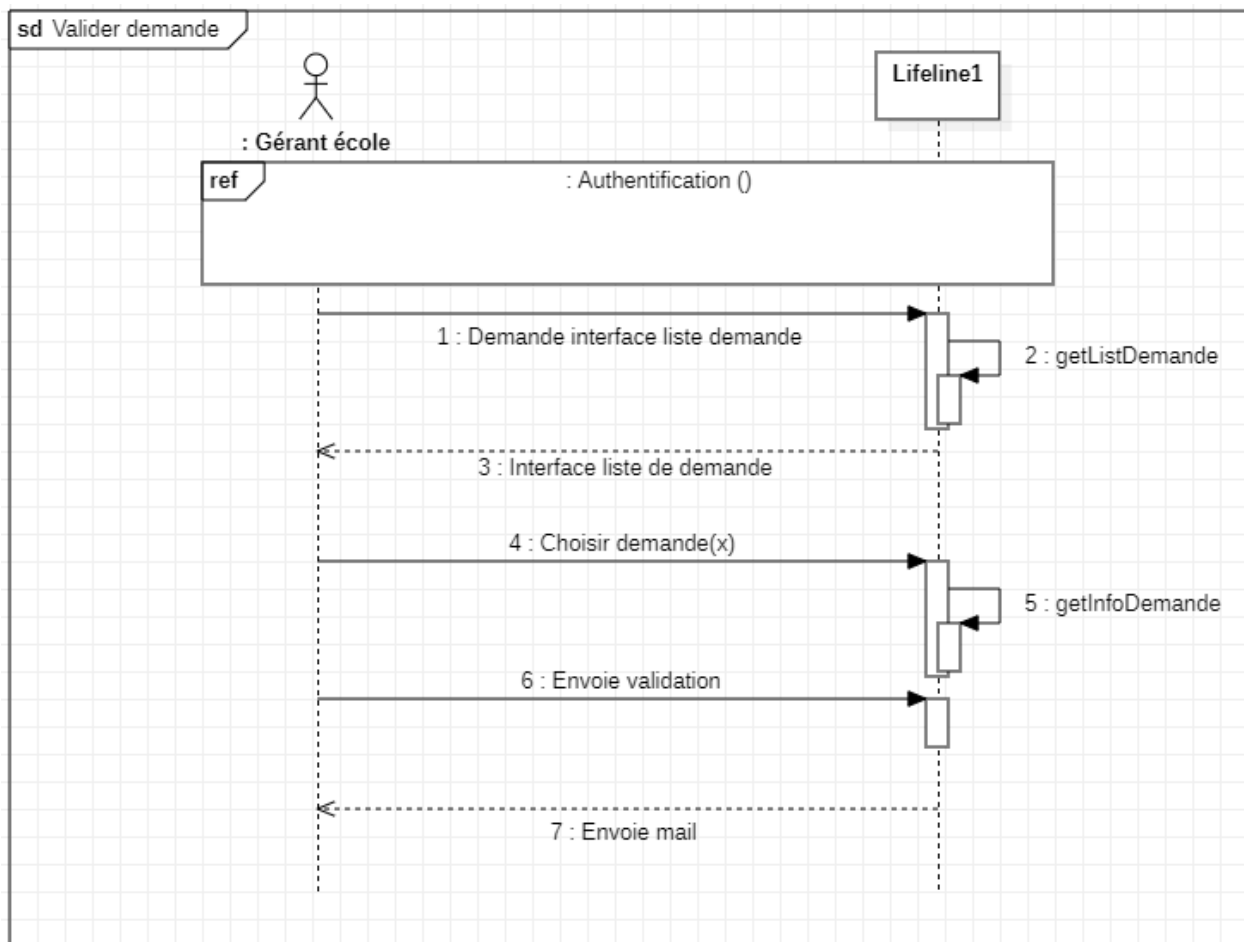


Figure 10: Séquence envoie de demande

## ➤ Séquence valider demande



*Figure 11: Séquence validation de demande*

## 2.2.2. Point de vue statique

### 2.2.2.1. Diagramme de classes

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

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

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

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

#### ➤ Diagramme de classe globale

# ERRATA

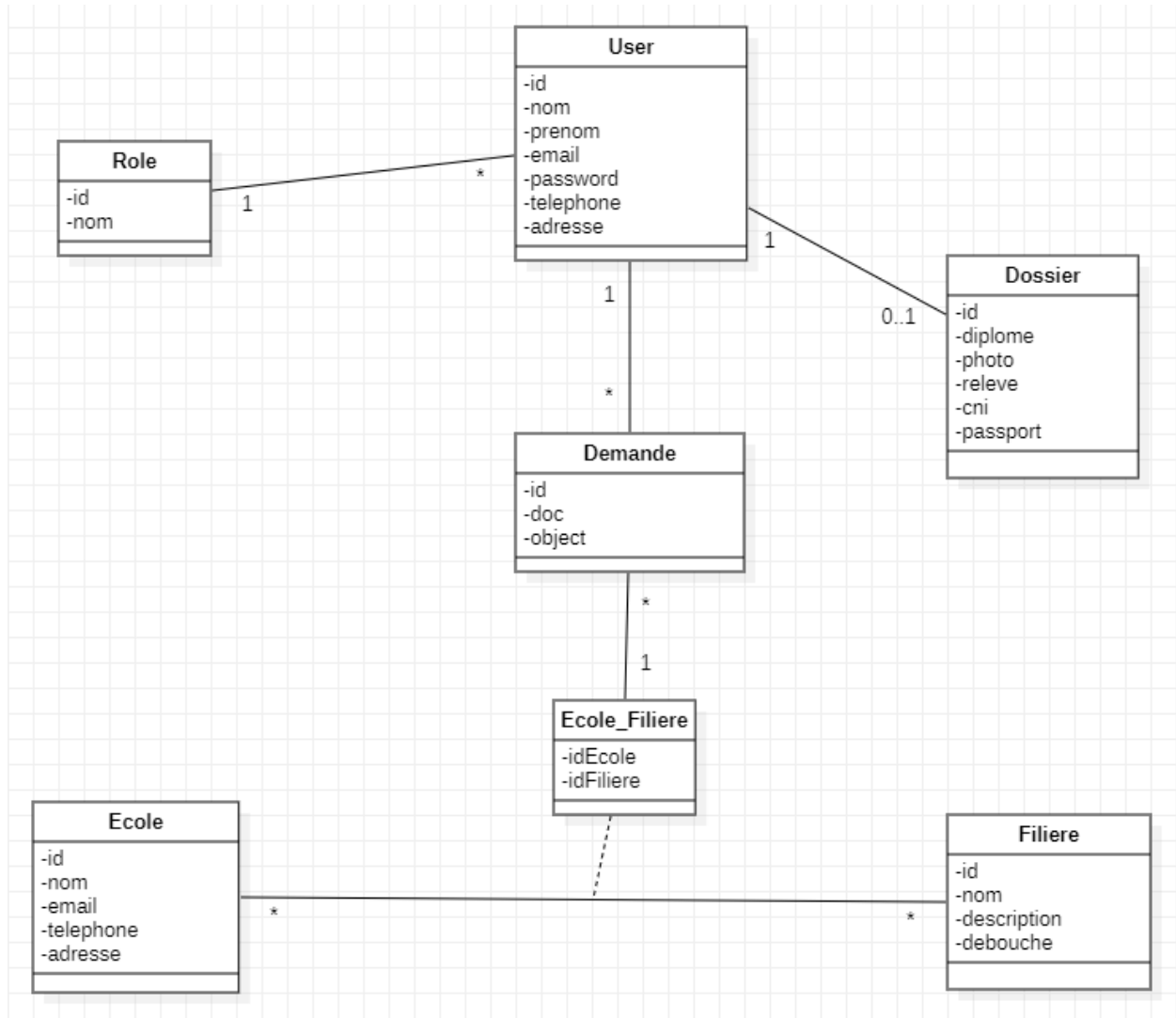


Figure 12: Diagramme de class globale

### 2.2.2.3. Diagramme de déploiement

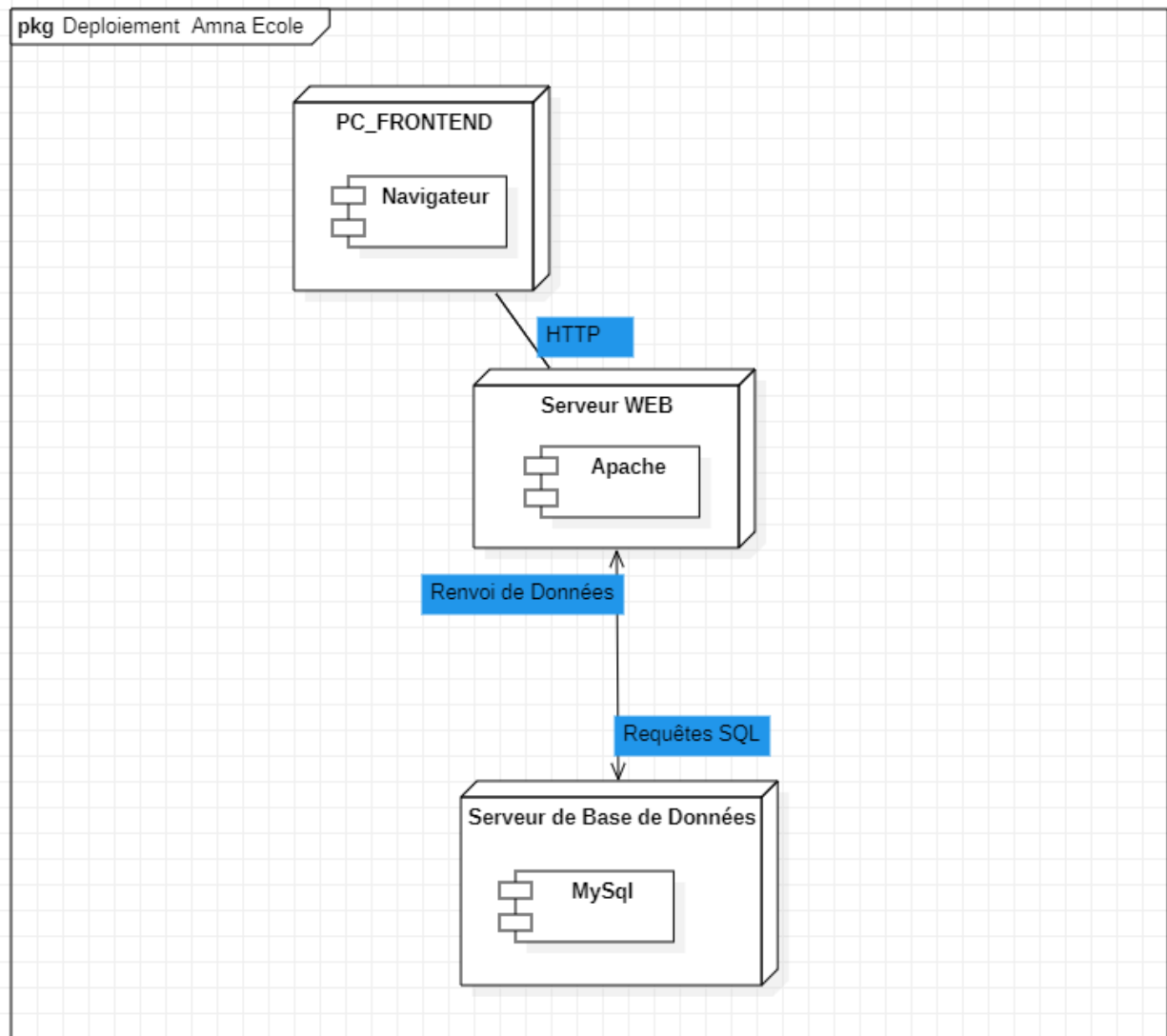


Figure 13: Diagramme de déploiement

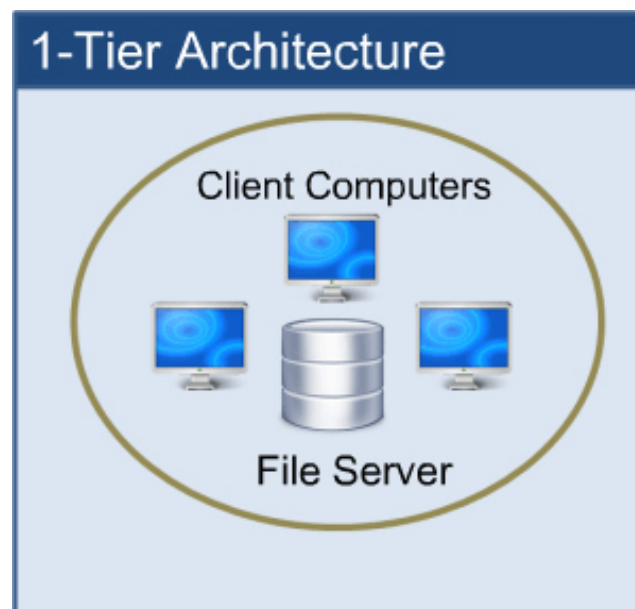
## **IMPLEMENTATION DU LOGICIEL**

### 3. Implémentation du logiciel

#### 3.1. Présentation des types d'architectures

##### 3.1.1. Architectures un-tiers

L'architecture à un niveau comporte la couche Présentation, la couche Entreprise et les couches Données au même niveau, c'est-à-dire au niveau Client. Comme son nom l'indique, toutes les couches et tous les composants sont disponibles sur la même machine. Lecteur MP3, MS Office, etc. sont quelques-uns des exemples d'architecture à un niveau. Pour stocker les données (en fonction de la couche de données), un système local ou un lecteur partagé est utilisé.



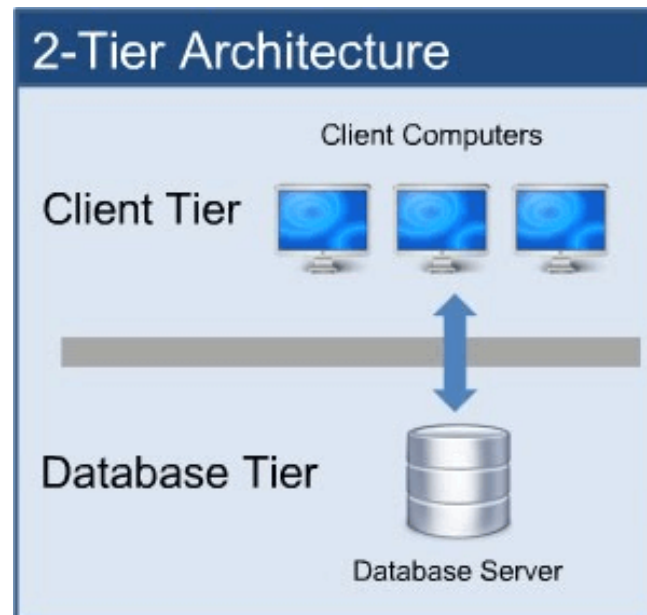
*Figure 14: Architectures un-tiers*

##### 3.1.2. Architecture deux-tiers

Dans ce type d'architecture, la couche client gère à la fois les couches Présentation et Application et le serveur gère la couche Base de données. L'architecture à deux niveaux est également connue sous le nom « d'application client-serveur ». Dans l'architecture à deux niveaux, la communication a lieu entre le client et le serveur. Le système client envoie la demande au système serveur et le système serveur traite la demande et renvoie la réponse au système client.

Les avantages d'une architecture client-serveur c'est que tout d'abord les ressources sont centralisées sur le serveur. Il est donc plus simple de gérer les ressources communes aux

utilisateurs comme la base de données par exemple. Ensuite, cette architecture est plus sécurisée étant donné que le client dispose de moins de point d'entrée pour accéder aux données.



*Figure 15: Architecture deux-tiers*

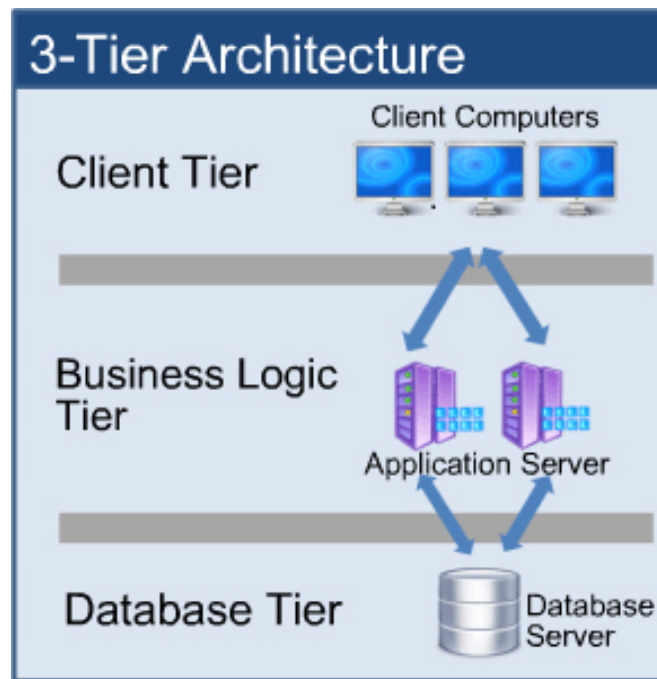
### 3.1.3. Architecture trois-tiers

Les trois couches principales sont séparées les unes des autres. La couche De présentation réside au niveau Client, la couche Application agit comme intermédiaire et se trouve au niveau Business et la couche Données est disponible au niveau Données. C'est une architecture très courante.

Les avantages d'une architecture 3-tiers sont nombreux. Tout d'abord cette architecture étant plus divisé permet d'avoir du point de vue du développement, une spécialisation des développeurs selon le niveau de l'application (un développeur front-end, back-end...)

Et enfin, cette architecture offre une flexibilité beaucoup plus importante que l'architecture 2- tiers. En effet, la portabilité du tiers serveur permet d'envisager une allocation et ou modification dynamique aux grés des besoins évolutifs au sein d'une entreprise.





*Figure 16: Architecture trois-tiers*

## 3.2. Réalisation de l'application

### 3.2.1. Langage de programmation

PHP (HyperText Preprocessor), plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. PHP est un langage impératif orienté objet. Il s'agit d'un langage de script interprété côté serveur.

Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.

Le logo considéré comme officiel du PHP est l'éléphant avec l'acronyme PHP sur son côté. Il est créé en 1998 par Vincent Pontier.

Le langage PHP fut créé en 1994 par Rasmus Lerdorf pour son site web. C'était à l'origine une bibliothèque logicielle écrite d'abord en langage Perl, puis en langage C. Il s'en servait pour conserver une trace des visiteurs qui venaient consulter son CV. PHP signifiait alors Personal Home Page tools. Le code source de PHP fut publié par son auteur en 1996. En 1997, deux étudiants, Andi Gutmans et Zeev Suraski, redéveloppèrent le cœur de PHP puis son moteur interne, aboutissant à ce qu'on appelle le Zend Engine.

Depuis les versions du langage se sont succédées, apportant leur lot d'améliorations et corrigeant les (nombreuses) failles de sécurité apparues en chemin. La version 5 de PHP, sortie en 2004, utilise Zend Engine 2 et introduit un véritable modèle objet, une gestion des erreurs fondée sur le modèle des exceptions, ainsi que des fonctionnalités de gestion pour les entreprises.

Même s'il peut être utilisé en ligne de commande, PHP est principalement associé à un serveur Web utilisant le protocole HTTP dans le cadre d'une architecture client/serveur.

Un serveur Web en architecture trois tiers est composé d'un système d'exploitation, un serveur HTTP, un langage serveur et enfin un système de gestion de base de données (SGBD), cela constituant une plate-forme.

Dans le cas de PHP comme langage serveur, les combinaisons les plus courantes sont celles d'une plateforme LAMP (pour Linux Apache MySQL PHP) et WAMP (Windows Apache MySQL PHP). Une plate-forme WAMP s'installe généralement par le biais d'un seul logiciel qui intègre Apache, MySQL et PHP, par exemple EasyPHP et WampServer. Il existe le même type de logiciels pour les plates-formes MAMP (Mac OS Apache MySQL PHP), à l'exemple du logiciel MAMP.

PHP appartient à la grande famille des descendants du C, dont les syntaxes sont très proches. En particulier, sa syntaxe et sa construction ressemblent à celles des langages Java et Perl, à ceci près que du code PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.

## ❖ Framework

Un Framework PHP est une plate-forme permettant de créer des applications web en PHP. Les Framework PHP fournissent des bibliothèques de code pour les fonctions les plus courantes, ce qui réduit la quantité de code original à écrire.

Pour notre travail nous avons choisi le framework **Laravel**. Laravel est présenté comme « Le framework PHP pour les artisans du web ». Il a été développé par Taylor Otwell, qui voulait un framework avec des éléments que CodeIgniter n'avait pas, comme l'authentification des utilisateurs.

## ➤ Spécifications rapides

Lancement : Juin 2011

Version actuelle : 9.0, publiée le 8 février 2022.

Exigences techniques :

- PHP >= 7.2.5 (ou utiliser Laravel Homestead)
- Composer installer
- Prise en charge des bases de données MySQL 5.6+, PostgreSQL 9.4+, SQLite 3.8.8+, SQL Server 2017+.

### ➤ Les avantages de Laravel

Il est facile de démarrer avec Laravel Homestead, un environnement de développement virtuel « fait pour vous ».

Laravel Homestead est une boîte Vagrant officielle, préemballée, qui vous offre un merveilleux environnement de développement sans vous obliger à installer PHP, un serveur web et tout autre logiciel de serveur sur votre machine locale. Plus besoin de vous inquiéter de la dégradation de votre système d'exploitation !

Si vous êtes un utilisateur de Mac, vous avez également le choix d'utiliser Laravel Valet comme environnement de développement. Soit dit en passant, Laravel Valet supporte Symfony, CakePHP 3, Slim et Zend, ainsi que WordPress.

Laravel utilise un moteur de templating appelé Blade. L'avantage qu'il présente par rapport aux autres moteurs de modèles est que vous pouvez utiliser PHP dans Blade, ce que vous ne pouvez pas faire avec les autres.

Packalyst, une collection de paquets Laravel, compte plus de 15 000 paquets que vous pouvez utiliser dans vos projets.

Laravel fournit une gamme de méthodes et de fonctionnalités de sécurité, couvrant les points suivants :

- Authentification
- Autorisation
- Vérification d'e-mail
- Cryptage
- Hachage

Réinitialisation du mot de passe

Eloquent ORM et Fluent Query Builder de Laravel protègent contre les attaques par injection SQL car ils utilisent la liaison de paramètres PDO. La protection contre la falsification de requêtes intersites (Cross-Site Request Forgery – CSRF), qui utilise un jeton de formulaire CSRF caché, est également activée par défaut.

L’outil de ligne de commande Artisan Console dont dispose Laravel accélère le développement en permettant aux développeurs d’automatiser les tâches répétitives et de générer rapidement du code squelette.

Lorsque nous avons effectué des tests d’évaluation des performances de PHP, Laravel était le plus rapide des frameworks PHP que nous avons essayés.

L’écosystème Laravel dispose de plusieurs outils utiles tels que Mix pour la compilation des actifs CSS et JS, et Socialite pour l’authentification OAuth.

Laravel bénéficie d’une large communauté de développeurs (comme WordPress).

### **3.2.2. Base de données**

Pour notre travail nous avons choisi le SGBD MySQL. MySQL est un SGBD gratuit, simple à utiliser et facilite l’hébergement. En effet la mise en place d’un serveur supportant MySQL est simple, car seul un serveur apache suffit. De ce fait presque tous les hébergeurs web offrent un hébergement MySQL.

MySQL est système de gestion de base de données relationnelle (SGBDR). C’est-à-dire un logiciel qui permet de gérer des bases de données, et donc gérer de grosses quantités d’information. Il utilise pour cela le langage SQL. Il s’agit d’un des SGBDR les plus connus et les plus utilisés (Wikipédia et adobe utilisent par exemple MySQL).

MySQL peut donc s’utiliser seul, mais la plupart du temps combiné à un autre langage de programmation : PHP, Java, Python, C++, et beaucoup d’autres.

#### **➤ Avantage**

- Solution très courante en hébergement public ;
- Rapide et Facile à utiliser ;
- Très bonne intégration dans l’environnement Apache ;
- Open source, bien que les critères de licence soient de plus en plus difficiles à supporter
- API diverse ;

- Connexion et sécurité ;
- Version cluster depuis la version 4 ;
- Ordonnanceur dès la version 5.1 ;
- Plusieurs moteurs de stockage adaptés à différentes problématiques, configurable au niveau table.

➤ **Inconvénients**

- Ne supporte qu'une faible partie des standards SQL-92 ;
- Support incomplet de triggers et procédures stockées ;
- Gestion de transaction avec les moteurs Falcon ou InnoDB uniquement ;
- Assez peu de richesse fonctionnelle ;
- Manque de robustesse avec de fortes volumétries ;
- Pas d'héritage de table ;
- Pas de vue matérialisée ;
- Pas de sauvegarde constante à chaud ;
- Cluster par clonage de base impact prépondérant sur la volumétrie.

### 3.2.3. Présentation de l'application

#### ➤ Page Admin

The screenshot shows the 'Ajouter une école' page in the admin interface. On the left is a dark sidebar with the 'amnaecole' logo and a navigation menu. The main content area is white and contains a form with the following fields: 'Nom de l'école', 'Courriel de l'école', 'Téléphone de l'école', and 'Adresse de l'école'. Below these is a section for 'Image de l'école' with a 'Choisir un fichier' button and the text 'Aucun fichier choisi'. At the bottom of the form is an 'Ajouter une école' button. The top of the page features a search bar, user icons, and a user profile dropdown labeled 'Ibrahima'.

*Figure 17: Interface administrateur*

#### ➤ Page d'accueil



*Figure 18: Interface d'accueil*

➤ Page des filières



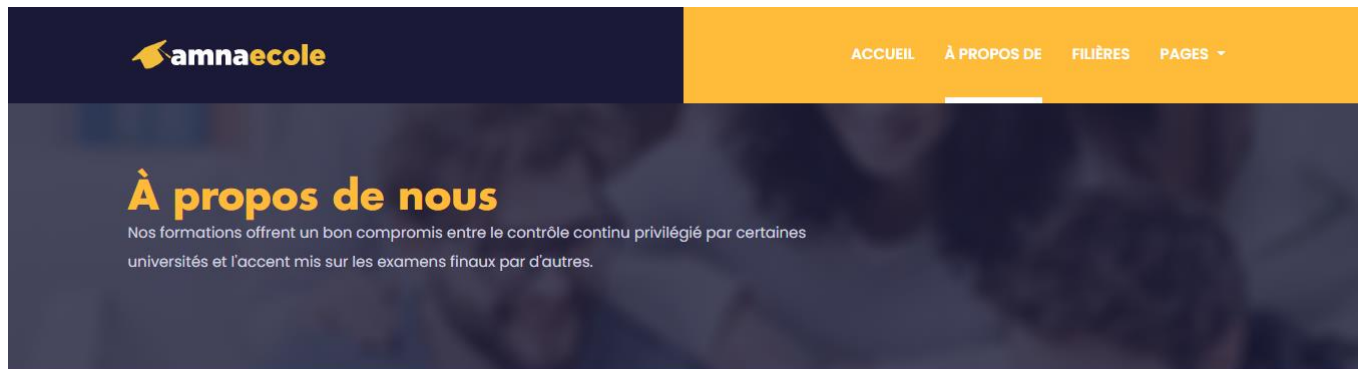
*Figure 19: Interface des filières*

➤ Page de consultation d'une filière



*Figure 20: Consultation d'une filière*

➤ Page de propos AMNA-ECOLE



## À PROPOS DE NOUS

AMNA ECOLE est une plateforme web de pré-inscription en ligne, qui permet aux nouveaux bacheliers de rechercher la filière et les établissements susceptibles de leur fournir la bonne formation et d'envoyer des demandes d'inscriptions aux instituts concernées.

*Figure 21: Interface renseignement*



## **Conclusion générale**

A travers ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation de notre plateforme de mise en place d'un système de préinscription universitaire pour des nouveaux bacheliers.

Afin de satisfaire les besoins des utilisateurs nous avons commencé la conception en utilisant le formalisme UML ensuite l'implémentation des requêtes SQL pour la manipulation des données et enfin l'utilisation du Framework Laravel sur Visual studio code.

Le travail que nous venons d'effectuer dans le cadre de l'obtention de la licence en informatique appliquée à la gestion des entreprise (IAGE), nous a permis d'approfondir nos connaissances acquises en programmation et en conception. Mais aussi d'autres aptitudes comme l'autonomie, la gestion de projet et du temps.

Cependant des perspectives d'améliorations de notre application restent envisageables telles que l'enrichissement du menu d'accueil en introduisant certaines fonctionnalités.

## Reference bibliographique

- ✚ Mémoire de fin cycle pour l'obtention de la licence professionnelle mise en place d'une plateforme de gestion de biens immobiliers (vente & location) pour l'entreprise Marly IMMO, **Auteur** : M. DIALLO Saifoulaye ;
- ✚ Uml2 de l'apprentissage à la pratique, **Auteur(s)** : Laurent Audibert, **Éditeur(s)** : Ellipses.

## Webographie

- + [https://fr.m.wikipedia.org/wiki/Diagramme\\_des\\_cas\\_d%27utilisation](https://fr.m.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation) visité le 20/08/2022
- + [https://fr.m.wikipedia.org/wiki/Diagramme\\_de\\_classes](https://fr.m.wikipedia.org/wiki/Diagramme_de_classes) visité le 01/10/2022
- + [https://fr.m.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://fr.m.wikipedia.org/wiki/Microsoft_SQL_Server) visité le 02/06/2022
- + [http://fr.abcarticulos.info/article/avantages-et-inconvnients-de-mysql\\_](http://fr.abcarticulos.info/article/avantages-et-inconvnients-de-mysql_), visité le 08/10/2021
- + <https://eaprovider.com/2022/besoin-non-fonctionnel-d-une-application-web/> visité le 07/10/2022
- + [Besoins fonctionnels & Besoins non fonctionnels - Savoir+](#), visité le 07/10/2022

## Table des matières

<b>Dédicace.....</b>	<b>i</b>
<b>Remerciements.....</b>	<b>ii</b>
<b>Avant-propos .....</b>	<b>iii</b>
<b>Sommaire .....</b>	<b>iv</b>
<b>Glossaire .....</b>	<b>v</b>
<b>Liste des figures .....</b>	<b>vi</b>
<b>Liste des tableaux .....</b>	<b>vii</b>
<b>Résumé.....</b>	<b>viii</b>
<b>Abstract .....</b>	<b>ix</b>
<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>1. Le cadre théorique et méthodologique.....</b>	<b>3</b>
<b>1.1. Le cadre théorique .....</b>	<b>3</b>
1.1.1. Présentation de la structure .....	3
1.1.2. Présentation du projet.....	3
1.1.2.1. Contexte .....	3
1.1.2.2. Problématique .....	3
1.1.2.3. Objectif .....	4
<b>1.2. Le cadre méthodologique.....</b>	<b>4</b>
1.2.1. Etude de l'existant.....	4
1.2.1.1. Critique de l'existant.....	5
1.2.1.2. Technique d'investigation.....	5
1.2.1.3. Méthode d'analyse et choix de la méthode.....	5
1.2.2. MERISE .....	5
1.2.3. UML .....	8

1.2.4.	Choix de la méthode.....	9
1.2.4.1.	Illustration de quelques diagrammes .....	10
<b>2.</b>	<b>Cadre organisationnel et conceptuel .....</b>	<b>13</b>
<b>2.1.</b>	<b>Cadre organisationnel.....</b>	<b>13</b>
2.1.1.	Identification des besoins .....	13
2.1.1.1.	Les besoins fonctionnels .....	13
2.1.1.2.	Les besoins non-fonctionnels.....	13
2.1.2.	Identifications des acteurs .....	14
<b>2.2.</b>	<b>Cadre conceptuel.....</b>	<b>14</b>
2.2.1.	Point de vue fonctionnel.....	14
2.2.1.1.	Diagramme de cas d'utilisation .....	14
2.2.1.2.	Description Textuelle de quelques cas d'utilisation .....	17
2.2.1.3.	Diagramme de séquence .....	20
2.2.2.	Point de vue statique .....	24
2.2.2.1.	Diagramme de classes.....	24
2.2.2.3.	Diagramme de déploiement .....	26
<b>3.</b>	<b>Implémentation du logiciel.....</b>	<b>28</b>
<b>3.1.</b>	<b>Présentation des types d'architectures .....</b>	<b>28</b>
3.1.1.	Architectures un-tiers .....	28
3.1.2.	Architecture deux-tiers.....	28
3.1.3.	Architecture trois-tiers.....	29
<b>3.2.</b>	<b>Réalisation de l'application .....</b>	<b>30</b>
3.2.1.	Langage de programmation.....	30
3.2.2.	Base de données .....	33
3.2.3.	Présentation de l'application .....	35
	<b>Conclusion générale .....</b>	<b>38</b>

<b>Reference bibliographique .....</b>	<b>39</b>
<b>Webographie .....</b>	<b>40</b>
<b>Table des matières .....</b>	<b>41</b>