

REPUBLIQUE DU NIGER
Fraternité-Travail-Progrès

**MINISTERE DE LA COMMUNICATION DES POSTES ET DE L'ECONOMIE
NUMERIQUE**

Ecole Supérieure des Communications Electroniques et de la poste (ESCEP-NIGER)

Ex EST Etablissement Public à Caractère Scientifique, Culturel et Technique créé par décret N°2023-226/PRN/MP/NTI du 02 mars 2023



**MEMOIRE DE FIN DE CYCLE POUR L'OBTENTION DU DIPLOME DE :
LICENCE EN GENIE LOGICIEL**

THEME :

**ETUDE ET MISE EN PLACE D'UNE APPLICATION POUR LA
GESTION ET L'AMORTISSEMENT DES IMMOBILISATIONS:**

cas de Gamma Informatique

Présenté Par :

M.MOHAMADOU SALIMOU Souradji

Encadré Par :

M.SADE Kassoum

Enseignant à l'ESCEP

Promotion : 2021-2024

DEDICACE

Je dédie ce travail à mes très chers parents, pour tout leur amour, encouragements et sacrifices, puisse Dieu leur prête une bonne santé et une longue vie afin que je puisse, à mon tour, les combler.

REMERCIEMENTS

Mes sincères remerciements vont au Directeur Général de Gamma Informatique **M.Ibra Kabo Mahaman** pour m'avoir accepté comme stagiaire au sein de l'entreprise.

Nous tenons également à remercier notre maître de stage **M.Yacouba Kouka** Informaticien au sein de Gamma pour son accueil chaleureux au niveau de son service, ses apports et conseils.

Nous tenons à présenter nos sincères remerciements et gratitude à l'égard de notre directeur de mémoire **M. Kassoum Sadé** pour nous avoir accompagné et soutenu tout au long de la réalisation de notre projet malgré ses multiples préoccupations.

Mes remerciements vont également à mon encadreur de stage **M. Adamou Ibrahim** Ingénieur en informatique de gestion à Gamma, pour sa patience et sa disponibilité pendant toute la durée de notre travail.

Nous saissons également cette occasion pour adresser nos profonds remerciements aux corps professoral et administratif de l'Ecole Supérieure des Communications Electroniques et de la Poste (**ESCEP-NIGER**) pour la richesse et la qualité de leur enseignement

Pour finir un grand merci à nos parents, à la famille, à nos amis et à tous ceux qui nous ont aidés de près ou de loin dans la réalisation de ce travail.

Sigles et Abréviations

MERISE : Méthode d'Etude Et De Réalisation Informatique Pour Les Systèmes d'Entreprise

ESCEP : Ecole Supérieure des Communications Electroniques et de la Poste

HTML : HyperText Markup Language

XML : eXtensible Markup Language

CSS : Cascading Style Sheets

JS : Java Script

POO : Programmation Orientée Objet

MCD : Modèle Conceptuel des Données

MLD : Modèle Logique des Données

MPD : Modèle Physique des Données

MCT : Modèle Conceptuel des Traitements

MVC : Model View Controller

MYSQL : My Structured Query Language

OMG : Object Management Group

OMT : Object Modeling Technique

OOSE : Object Oriented Software Engineering

SGBDR : Système de Gestion de Base de Données Relationnelles

2TUP : Two Track Unified Process

UP : Unified Process

UML : Unified Modeling Language

USB : Universal Serial Bus

VS Code : Visual Studio Code

PHP : HyperText Preprocessor

RUP : Rational Unified Process

Listes des figures

Figure 1 : organigramme de la structure	9
Figure 2 : Figure de développement en Y	16
Figure 3: Diagrammes UML 1.4	18
Figure 4:Organigramme UML version 2.5.X	20
Figure 5 : Figure MVC.....	28
Figure 6 : Figure client /serveur.....	30
Figure 7:Diagramme de cas d'utilisation pour l'administrateur	37
Figure 8 : Diagramme de cas d'utilisation pour le comptable	38
Figure 9 : Diagramme d'activité du cas d'utilisation « Authentification ».....	43
Figure 10 : Diagramme d'activité du cas d'utilisation « ajouter <i>immobilisation</i> ».....	43
Figure 11 : Diagramme d'activité « modifier immobilisation »	44
Figure 12 : Diagramme d'activité « ajouter une Cession »	45
Figure 13 : Diagramme de séquence du cas d'utilisation « Ajouter immobilisation »	46
Figure 14 : Diagramme de séquence pour le cas d'utilisation « calcul amortissement ».	47
Figure 15 : Diagramme de séquence « céder immobilisation »	48
Figure 16 : Diagramme globale su système.....	49
Figure 17 : Diagramme de déploiement	50
Figure 18 : Page d'authentification.....	55
Figure 19 : Liste des immobilisations cédées	55
Figure 20 : Formulaire permettant d'ajouter une immobilisation	56
Figure 21 : Immobilisations concernées pour une annexe a doté	56

Listes des tableaux

Tableau 1 : Tableau comparative entre Merise et UP	15
Tableau 2:Tableau des acteurs des systèmes	34
Tableau 3:Tableau des besoins fonctionnels	35
Tableau 4: Description textuelle du cas d'utilisation « authentification »	39
Tableau 5 : Description textuelle « d'ajout immobilisation »	40
Tableau 6 : Description textuelle de « calcul d'amortissement ».....	41
Tableau 7 : Description textuelle du cas d'utilisation « ajout compter utilisateur»	42

Table des matières

DEDICACE	i
REMERCIEMENTS	ii
Sigles et Abréviations	iii
Listes des figures.....	v
Listes des tableaux.....	vi
Table des matières	ii
INTRODUCTION	1
PREMIERE PARTIE : CADRE THEORIQUE ET METHODOLOGIQUE	1
CHAPITRE I : CADRE THEORIQUE	3
I. Problématique.....	3
II. Objectifs de recherche	3
II.1 Objectif général	3
II.2 Objectifs spécifiques.....	3
III. Hypothèse de recherche	4
III.1 Hypothèse générale	4
III.2 Hypothèses spécifiques	4
IV. Intérêt général	4
CHAPITRE II : CADRE METHODOLOGIQUE	5
I. Cadre d'étude	5
II. Délimitation du champ d'étude.....	5
III. Technique d'investigation.....	5
IV. Difficultés rencontrées	5

DEUXIEME PARTIE : CADRE ORGANISATIONNEL ET CONCEPTUEL	6
CHAPITRE I : CADRE ORGANISATIONNEL.....	7
I. Présentation	7
II. Les valeurs de Gamma Informatique.....	7
III. Les pôles de compétences de Gamma Informatique	8
IV. Les moyens humains	8
V. Les moyens matériels	8
VII. Organigramme	9
CHAPITRE II : CADRE CONCEPTUEL	10
I. Définition des mots et concepts relatifs au thème.....	10
I.1 Immobilisations	10
I.2 Amortissement.....	10
I.3 Cession	11
I.4 Dotation.....	11
II. Méthode d'analyse et conception	11
II.1 MERISE	12
II.1.1 les cycles de Merise.....	12
II.2 Le processus Unifié (PU).....	13
II.2.1 Caractéristiques de processus unifié.....	13
II.2.2 Les phases du cycle de vie du processus unifié (UP)	14
II.2.3 Tableau comparative entre Merise et UP.....	14
II.2.4 Choix de la méthode d'analyse.....	15
III. Langage de modélisation UML	17
III. 1.1 Version 1.4 de UML.....	17
III.1.2 Version 2.5 de UML.....	19
IV. Langage de programmation	20
IV.1 Langage de programmation de procédurale	20

IV.2 Langage de programmation orienté objet	21
IV.2.1 Etude comparative	21
IV.2.2 Choix et justification du langage de programmation	23
V. Système Gestion de Bases de Données Relationnelles (SGBDR)	23
V.1 Etude comparative.....	24
IV.2 Choix et justification.....	25
VI. Technologie de développement	26
V.1 Technologie Web.....	26
V.1.1 Framework	26
V.2 Technologie mobile	27
V.3 Technologie Desktop.....	27
VII. Architecture logicielle	27
VI.1 L'architecture Modèle Vue Contrôleur (MVC)	27
VI.2 architecture client/serveur	29
VI.3 choix de l'architecture	30
TROISIEME PARTIE : CADRE ANALYTIQUE	31
CHAPITRE I : ANALYSE ET CONCEPTION.....	32
I. ANALYSE.....	32
I.1 Etude de l'existant	32
I.1.1 Analyse de l'existant	32
I. 1.2 Critique de l'existant	32
I.2 Solution envisagées.....	33
I.3 Etude des besoins	33
I.3.1 Identification des acteurs	33
I.3.2 Besoins fonctionnels	34
I.3.4 Besoins non fonctionnels.....	35
II. CONCEPTION.....	36

II.1 Diagramme de Cas d'utilisation.....	36
II.1.1 Les relations entre les cas d'utilisation.....	36
II.1.2 Représentation du diagramme cas d'utilisation	36
II.2 Diagramme d'activités	42
II.2.1 Représentation du diagramme d'activité de quelque cas	42
II.3 Diagramme de séquence	46
II.3.1 Représentation de quelques diagrammes de séquence	46
II.4 Diagramme de classe	48
II.4.1 Représentation du diagramme de classe	49
II.5 Diagramme de déploiement.....	49
Synthèse.....	50
CHAPITRES II : REALISATION	51
I. Outils et librairies utilisés	51
I.1 Outils de modélisation	51
i. Power AMC.....	51
I.2 Outils de développement	51
i. Visual Studio Code	51
ii. HTML et CSS	51
iii. PHP.....	52
iv. WampServer.....	52
v. MySQL	52
I.3 Librairies	52
i. Bootstrap	52
ii. JavaScript	53
iii. JQuery.....	53
iv. Ajax	53
II. Politique de confidentialité	53

II.1 Sécurisation contre les attaques	54
II.2 La sécurisation contre les virus	54
II.3 Confidentialité des données	54
III. Quelques capture d'écran.....	55
Conclusion	57
Bibliographie	58
Webographie	59

INTRODUCTION

Durant ces dernières années, l'informatique s'est imposée d'une manière très impressionnante dans les entreprises, cela est dû à son apport extraordinaire dans le domaine de l'automatisation des systèmes d'information et de la gestion des bases de données.

En effet, depuis l'avènement de l'informatique, les entreprises aspirent à améliorer l'efficacité et la fiabilité de la gestion de leurs structures internes. Cependant, la gestion des immobilisations, qu'elles soient corporelles ou incorporelles, ainsi que l'amortissement de ces actifs, peuvent devenir des tâches complexes et fastidieuses, nécessitant une approche méthodique et une attention particulière.

Dans ce contexte spécifique, notre projet de mémoire se positionne comme une étude approfondie visant à explorer et à mettre en œuvre des solutions innovantes pour la gestion des immobilisations et de l'amortissement au sein de Gamma Informatique. Dans le cadre de l'obtention du diplôme de licence en génie logiciel, nous proposons d'analyser et de développer une application dédiée à la gestion de ces actifs au sein de l'entreprise. Ainsi, le thème de notre étude est défini comme suit : « **Étude et mise en place d'une application pour la gestion et l'amortissement des immobilisations : cas de Gamma Informatique** »

Ce mémoire se structure en trois parties fondamentales, chacune contribuant de manière significative à la réalisation de notre projet :

- ❖ La première partie, « **Cadre Théorique et Méthodologique** », sera dédiée à l'exploration des fondements théoriques et méthodologiques de la gestion des immobilisations et de l'amortissement.
- ❖ La deuxième partie, « **Cadre Organisationnel et Conceptuel** », abordera la structure organisationnelle de Gamma Informatique, ainsi que les différentes méthodes d'analyse et de conception utilisées pour le développement de notre application.
- ❖ Enfin, la troisième partie, « **Cadre Analytique** », se concentrera sur l'analyse et la conception détaillées de notre application de gestion des immobilisations et de l'amortissement au sein de Gamma Informatique. Nous y présenterons les outils utilisés pour son développement, ainsi que les mesures de sécurisation mises en place, avant de fournir quelques exemples concrets d'interfaces utilisateur générées.

PREMIÈRE PARTIE : Cadre Théorique et Méthodologique

CHAPITRE I : CADRE THEORIQUE

I. Problématique

La gestion des immobilisations et des amortissements constitue un enjeu crucial pour toute entreprise, c'est le cas de Gamma Informatique, car elle affecte directement sa santé financière, sa performance opérationnelle et sa conformité réglementaire. Ainsi, grâce à nos questions posées au personnel, nous avons relevé quelques difficultés auxquelles ce domaine fait face, telles que :

- la complexité des processus et la recherche d'actifs ;
- la gestion manuelle des actifs ;
- la conformité aux normes comptables.

Tout au long de notre travail, nous serons amenés à apporter des réponses aux questions suivantes :

- Quel système faut-il mettre en place pour gérer les immobilisations au sein de Gamma Informatique sans être lié à un logiciel comptable ?
- Comment l'application va-t-elle faciliter le suivi des immobilisations ?
- Comment l'application va-t-elle permettre de faire des plans d'amortissement ?
- Comment l'application va-t-elle permettre de céder et de faire des dotations pour les immobilisations ?

II. Objectifs de recherche

II.1 Objectif général

L'objectif général de notre projet est de concevoir, développer et mettre en place une application pour la gestion et l'amortissement des immobilisations au sein de Gamma Informatique.

II.2 Objectifs spécifiques

La réalisation de notre application permettra de :

- Stocker les informations dans une base de données ;
- La rapidité dans la recherche des actifs de l'entreprise ;
- Faciliter le suivi des actifs

- Calculer et faire des plans d'amortissements
- Faciliter les cessions et les dotations

III. Hypothèse de recherche

III.1 Hypothèse générale

L'intégration d'une application de gestion des immobilisations et de l'amortissement au sein de Gamma Informatique permettra d'optimiser la gestion de ses ressources matérielles, d'améliorer la visibilité et le suivi de ses actifs.

III.2 Hypothèses spécifiques

L'application de gestion des immobilisations et de l'amortissement permettra de :

- Réduire les erreurs et les lacunes dans le suivi des actifs ;
- Automatiser les processus de gestion des immobilisations et de l'amortissement grâce à l'application ;
- Sécuriser les données sensibles liées aux immobilisations et aux amortissements, garantissant ainsi la confidentialité et l'intégrité des informations ;
- Suivre l'ensemble des biens acquis par l'entreprise ;
- Enregistrer les informations relatives à ces biens ;
- Calculer les amortissements conformément aux règles comptables en vigueur.

IV. Intérêt général

Le projet va nous permettre de mettre en pratique les connaissances théoriques et pratiques acquises au cours de notre formation, de développer un esprit d'analyse et de conception et de nous familiariser avec les outils de programmation.

Notre étude permettra ainsi de faciliter et d'améliorer la qualité du travail au sein de Gamma Informatique sur le plan de la gestion des immobilisations et amortissements

CHAPITRE II : CADRE METHODOLOGIQUE

I. Cadre d'étude

Notre étude se base sur la conception et la mise en place d'une application de gestion des immobilisations ainsi que de leurs amortissements pour optimiser ces dernières au sein de Gamma Informatique. Ce document s'inscrit dans le cadre de l'obtention du diplôme en Génie Logiciel. Pour ce faire, nous avons effectué un stage de trois (3) mois au sein de Gamma Informatique.

II. Délimitation du champ d'étude

Pour mener à bien notre projet, et comme tout travail scientifique, nous avons jugé utile de délimiter notre travail dans l'espace et dans le temps.

Ainsi dans le temps, notre étude se portera sur une période de trois (3) mois, période durant laquelle nous recenserons les données nécessaires pour une meilleure analyse et enfin aboutir à la réalisation.

Dans l'espace, nous allons nous intéresser aux processus d'inventaire, de gestion et de suivis des immobilisations.

III. Technique d'investigation

Pour la réalisation de ce travail, nous avons eu recours à plusieurs techniques, parmi lesquelles:

- ❖ Echanger avec le maître du projet et autres personnes impliquées ;
- ❖ La consultation de certains manuels ayant un lien avec notre domaine de recherche comme les ouvrages et les mémoires.

IV. Difficultés rencontrées

L'une des principales difficultés que nous avons rencontrées au cours de la réalisation du présent projet est le temps nécessaire pour comprendre le processus de gestion des immobilisations, ainsi que la mauvaise qualité de la connexion, voire presque un manque de connexion Internet au sein de la structure pendant nos recherches.

DEUXIEME PARTIE : CADRE ORGANISATIONNEL ET CONCEPTUEL

CHAPITRE I : CADRE ORGANISATIONNEL

I. Présentation

GAMMA NIGER est une Société Anonyme avec un capital de 102.000.000 F CFA, créée en novembre 1989. Située en plein centre de Niamey, au Niger, dans le quartier des affaires, GAMMA œuvre constamment pour le développement des établissements publics et privés, tant au niveau informatique qu'organisationnel. Grâce à son réseau d'affaires et à ses relations internationales, GAMMA NIGER est en mesure de répondre aux besoins et aux exigences de ses clients dans les meilleurs délais.

Depuis 1997, GAMMA NIGER a souscrit auprès de la SNAR-LEYMA un contrat d'assurance couvrant la Responsabilité Civile Professionnelle ; les clients sont ainsi sécurisés quant aux risques éventuels qui pourraient survenir pendant ou à la suite de nos prestations.

Gamma Niger, bien connue pour la fourniture de solutions organisationnelles et informatiques clé-en-main, est une société qui intervient dans plusieurs domaines et assure des prestations dans plusieurs pays. À travers ses activités, elle met à la disposition de ses clients ses compétences afin de les aider à prendre des décisions judicieuses dans le choix des démarches organisationnelles ainsi que des équipements informatiques. Pour satisfaire les exigences de sa clientèle, GAMMA s'est dotée d'un système de management de la qualité conforme au référentiel international ISO 9001:2008.

II. Les valeurs de Gamma Informatique

- PARTAGER LA PASSION

Les collaborateurs sont épris par le métier et généreux dans le transfert de savoir

- TOUT CLIENT EST UNIQUE

Personnaliser, c'est tout d'abord comprendre avant de répondre

- SOUTENIR LA CRÉATIVITÉ

La création et l'innovation sont au cœur de leurs processus de fonctionnement

- SENS DU DÉTAIL

Ils sont toujours à la quête de l'excellence et du travail bien fait

- PROXIMITÉ ET RIGUEUR

Repère commun: écoute, disponibilité et un réel support

III. Les pôles de compétences de Gamma Informatique

Gamma couvre plusieurs métiers

- La vente et maintenance de matériel informatique
- Les études
- La formation et certification

IV. Les moyens humains

GAMMA est managé par un Comité de Direction composé d'un directeur général, de directeurs métiers et de directeurs supports. Chaque direction, en fonction de sa spécialité, dispose des ressources nécessaires qualifiées, tant au niveau des cadres que des techniciens.

Aujourd’hui Gamma dispose de :

- **13 Cadres**
- **8 techniciens**
- **9 Agents**

Soit un taux d’encadrement de 43 %

V. Les moyens matériels

GAMMA dispose de deux (2) sites, à savoir la direction générale et l’annexe, tous situés en centre-ville, ce qui lui confère une bonne position stratégique.

Comme moyens techniques, les agents disposent de véhicules de travail ainsi que de tout l’appareillage et l’outillage professionnels nécessaires pour bien servir ses clients.

VII. Organigramme

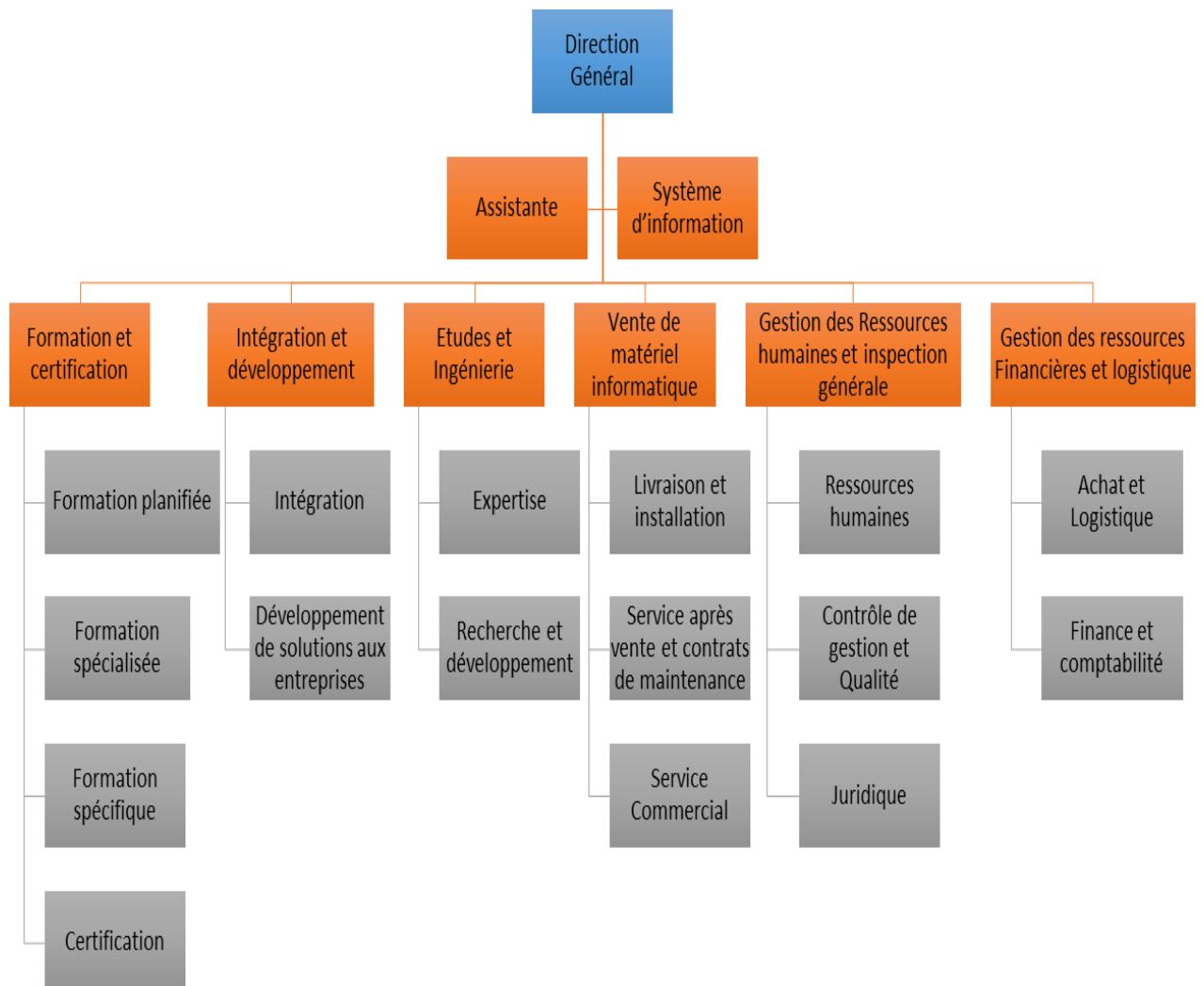


Figure 1 : organigramme de la structure

CHAPITRE II : CADRE CONCEPTUEL

Notre travail sur ce chapitre consiste à parler d'abord de quelques concepts relatifs à notre thème, des différentes méthodes d'analyse et de conception, faire une comparaison entre ces méthodes afin de choisir la méthode à utiliser pour la réalisation de notre projet , parler des langages de programmation , des bases de données des technologies de développement et d'architecture .

I. Définition des mots et concepts relatifs au thème

I.1 Immobilisations

Les immobilisations sont généralement regroupées en trois grandes catégories :

- Les immobilisations incorporelles : ce sont des actifs intangibles, ne pouvant ni être touchés ;
- Les immobilisations corporelles : ce sont des actifs tangibles, concrets, utilisés dans l'entreprise ;
- Les immobilisations financières : également connues sous le nom d'actifs financiers non courants, ce sont des investissements à long terme dans d'autres entreprises ou entités qui ne sont pas détenus à des fins de négociation immédiate ou de liquidité rapide.

I.2 Amortissement

L'amortissement est un processus comptable qui consiste à répartir le coût d'un actif sur sa durée de vie utile. En d'autres termes, c'est la manière dont une entreprise étaie le coût d'un actif sur plusieurs périodes comptables pour refléter sa dépréciation progressive et son usure au fil du temps.

Il existe généralement deux types d'amortissements qui sont :

- **L'amortissement linéaire** : qui consiste à répartir de manière égale les dépréciations d'une immobilisation sur sa durée de vie réelle. Il est également appelé « amortissement constant ».

Amortissement linéaire de la 1ère et dernière année = Valeur d'acquisition x Taux d'amortissement x Prorata temporis / 360

Puis, au titre des années « pleines » :

Amortissement linéaire = Valeur d'acquisition x Taux d'amortissement

- **L'amortissement dégressif :** il n'a que très rarement une justification « économique ». Il s'agit, en réalité, d'un avantage fiscal accordé aux entreprises. Il permet d'amortir très fortement une immobilisation au début de son utilisation.

Formule :

Amortissement Dégressif=Valeur d'acquisition*Taux d'amortissement dégressif Avec

Taux d'amortissement dégressif=Taux linéaire* Coefficient multiplicateur

Taux linéaire = (1/durée de vie)*100

I.3 Cession

Il s'agit tout simplement du fait de vendre ou mettre en hors service ce bien qui a été utilisé par votre entreprise. La cession d'immobilisation conduit donc à sortir l'immobilisation de l'actif de votre bilan. Elle ne fait alors plus partie du patrimoine de l'entreprise.

I.4 Dotation

En comptabilité, la dotation désigne une somme d'argent prévue et affectée à une charge ou une provision dans les états financiers de l'entreprise. Il s'agit d'un montant qui est inscrit au compte de résultat de l'entreprise pour représenter une charge future ou une diminution de la valeur d'un actif

II. Méthode d'analyse et conception

Une méthode d'analyse et de conception, dans le contexte du développement de logiciels ou de systèmes, est une approche systématique pour comprendre les besoins des utilisateurs, concevoir des solutions pour répondre à ces besoins, et ensuite réaliser ces solutions de manière efficace et robuste.

La phase d'analyse dans le processus de développement logiciel est une étape cruciale où les besoins, les exigences et les contraintes du système à concevoir sont étudiés en détail. Cette phase vise à comprendre et à spécifier ce que le système doit faire sans se préoccuper de la manière dont il sera réalisé techniquement.

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

De ce fait, il existe plusieurs méthodes d'analyse et de conception telles que : MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise), le processus unifié(UP), Agile, etc.

Dans le cadre de notre travail nous allons définir et faire une étude comparative entre MERISE et UP à fin de faire un choix.

II.1 MERISE

MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) est une méthode d'analyse et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes : schéma directeur, étude préalable, étude détaillée et réalisation.

Elle possède plusieurs modèles répartis sur trois niveaux : le niveau conceptuel, le niveau logique ou organisationnel, et le niveau physique.

Les schémas les plus couramment utilisés sont le MCD (Modèle Conceptuel de Données), le MLD (Modèle Logique de Données) et le MPD (Modèle Physique de Données), souvent représentés sous la forme de diagrammes événement-résultat pour représenter la dynamique des processus.

II.1.1 les cycles de Merise

MERISE présente dans sa démarche d'analyse trois cycles fondamentaux : le cycle d'abstraction, le cycle de vie (de développement), le cycle de décision.

1. Le cycle de vie : comporte trois grandes périodes :

- La conception : période d'étude de l'existant puis du système à mettre en place,
- La réalisation : recouvre la mise en œuvre et l'exploitation,
- La maintenance : devra permettre au système d'évoluer et de s'adapter aux modifications de l'environnement et aux nouveaux objectifs pendant une certaine durée de vie et ensuite il devra laisser la place à un nouveau système.

2. Le cycle d'abstraction : concerne le système de spécification d'un SI (Système d'information):

- La mémoire du SI est décrite sur le plan conceptuel, puis logique et enfin physique.
- Les processus de traitement sont décrits sur le plan conceptuel, puis organisationnel et enfin opérationnel.

Chaque couche est décrite sous la forme d'un modèle. Lorsque les paramètres des couches inférieures évoluent, la couche décrite n'est pas altérée, et elle ne le sera que si ses propres paramètres seront modifiés.

Chaque modèle est décrit à travers un formalisme reposant sur des règles et des principes, un vocabulaire et une syntaxe. Des règles de transition permettent de passer plus ou moins automatiquement d'un modèle à un autre.

3. Le cycle de décision : Tout au long de l'étude et de la maintenance, des décisions sont à prendre, très générales d'abord puis de plus en plus détaillées.

Les décisions globales sont prises par la direction générale mais, à chaque niveau, chacun doit être consulté

II.2 Le processus Unifié (PU)

Le processus unifié (PU), ou « unified process (UP) » en anglais est une famille de méthodes de développement de logiciels orientés objets. Il se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML. Il définit un processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement composés d'une phase de création, d'une phase d'élaboration, d'une phase de construction et d'une phase de transition, comprenant chacune plusieurs itérations

II.2.1 Caractéristiques de processus unifié

La méthode du processus unifié possède les caractéristiques suivantes :

- ❖ **Itérative et Incrémentale** : itérative, parce que chaque phase de la démarche UP comprend des itérations ; incrémentale puisqu'il incrémente le projet par itération c'est à dire que le logiciel et le modèle évoluent suivant des incréments ;
- ❖ **Piloté par les cas d'utilisation** : les besoins des utilisateurs permettent de définir les cas d'utilisations aboutissant à une création de série UML.

L'utilisateur représente une personne dialoguant avec le système et l'objectif est d'essayer de comprendre ses désirs et ses besoins. Les cas d'utilisations permettent d'illustrer les besoins fonctionnels afin de garantir une meilleure qualité du futur logiciel ;

- ❖ **Centré sur l'architecture** : cette architecture doit prévoir la réalisation de tous les cas d'utilisations. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit ;
- ❖ **Orienté par la réduction des risques** : UP contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives.

II.2.2 Les phases du cycle de vie du processus unifié (UP)

Le processus unifié est cyclique : il vise par une succession de projets à fournir d'abord une version viable d'un produit puis des versions publiables successives (« release » en anglais).

Chaque projet a un cycle de vie en quatre phases, chacune subdivisée en plusieurs itérations :

- ❖ Une phase de **création** ou de cadrage (« *Inception* » en anglais) vise à définir le produit et les objectifs du projet ;
- ❖ Une phase d'**élaboration** vise à clarifier les exigences, à définir l'architecture du produit et à en valider la faisabilité. Cette phase prévoit ainsi la mise en œuvre d'une version exécutable constituée d'un squelette du système et démontrant les principaux éléments architecturaux ;
- ❖ Une phase de **construction** vise à construire et à mettre en œuvre le produit et les livrables associés ;
- ❖ Une phase de **transition** vise à livrer, diffuser ou déployer le produit de sorte qu'il soit prêt à être utilisé. Cette phase inclut la formation des utilisateurs si nécessaire.

Il existe donc un certain nombre de méthodes issues d'UP comme **RUP** (Rational Unified Process) et **2TUP** (Two Track Unified Process).

II.2.3 Tableau comparative entre Merise et UP

Méthodes	Point Fort	Point Faible
MERISE	<ul style="list-style-type: none"> • Approche Systémique ; • Séparation des données et des traitements ; • Concept peut nombreux et simple ; 	<ul style="list-style-type: none"> • La méthode s'arrête au niveau organisationnel pour les traitements et les données. • Elle ne s'occupe pas de l'interface utilisateur.
Processus Unifié (UP)	<ul style="list-style-type: none"> • Itérative ; • Spécifie le dialogue entre différents intervenants ; • Propose des modèles pour des projets. 	<ul style="list-style-type: none"> • Coûteux à personnaliser; • Peu de place pour le code et la technologie ; Cible les projets de plus de 10 personnes

Tableau 1 : Tableau comparative entre Merise et UP

II.2.4 Choix de la méthode d'analyse

Pour la réalisation de notre projet, notre choix se porte sur le **Two Track Unified Process** (2TUP) puisque c'est un processus de développement qui implémente le **Processus Unifié (UP)**.

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages échangés entre les acteurs et le système, à produire le cahier des charges et à modéliser le contexte

Le processus s'articule ensuite autour de trois phases essentielles :

- Une branche technique ;
- Une branche fonctionnelle ;

- Une phase de réalisation.

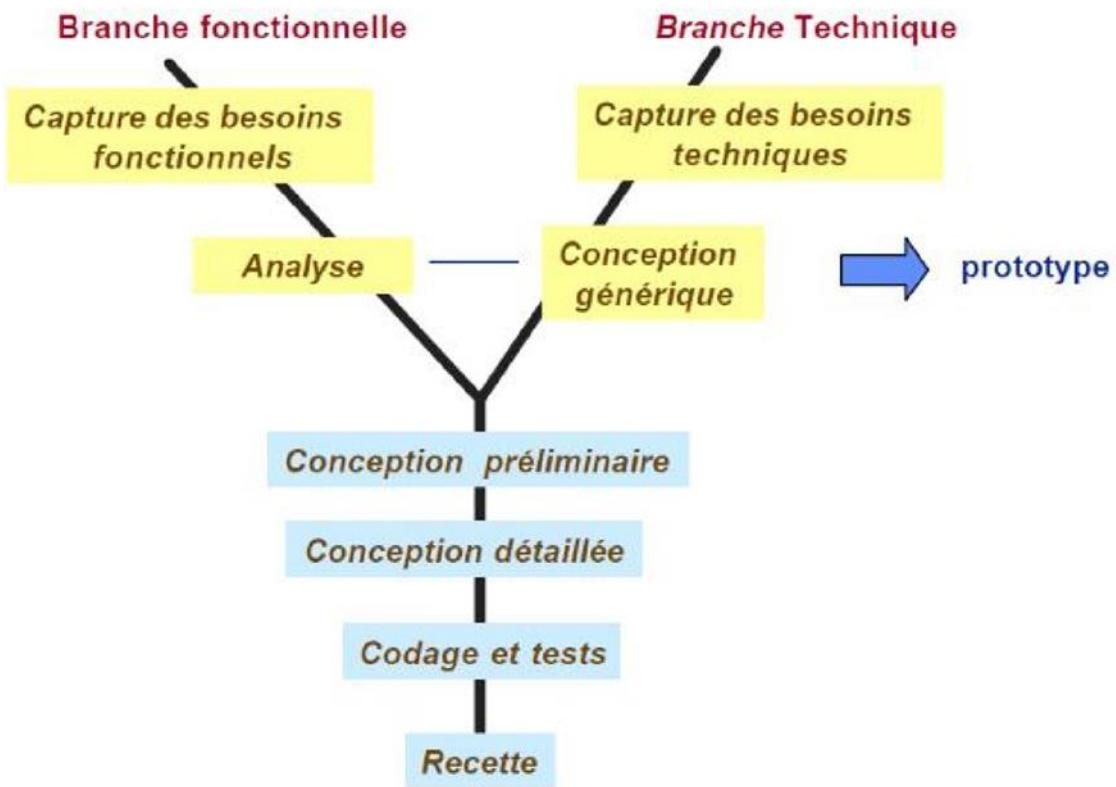


Figure 2 : Figure de développement en Y

- ✓ La branche fonctionnelle capitalise la connaissance du métier de l'entreprise. Cette branche capture les besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.
- ✓ La branche technique capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.
- ✓ La phase de réalisation consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

III. Langage de modélisation UML

Le **Langage de Modélisation Unifié**, de l'anglais Unified Modeling Language (**UML**), est une démarche orientée objet. En effet, il s'agit d'un langage de modélisation graphique à base de pictogrammes (diagrammes) conçus pour fournir une méthode normalisée afin de visualiser la conception d'un système, couramment utilisé en développement de logiciel et en conception orientée objet. UML est né de la fusion des trois (3) langages de modélisation objet à savoir **Booch**, **OMT** et **OOSE** des fameux scientifiques, respectivement, Grady Booch, James Rumbaugh et Ivar Jacobson.

UML est à présent un standard adopté par l'Object Management Group (OMG)

UML présente plusieurs versions dont entre autres nous pouvons citer la version 2.5.1 et la version 1.4

III. 1.1 Version 1.4 de UML

UML 1.4 propose **neuf diagrammes**.

Chacun décrit une **partie du système** ou décrit le système selon un **point de vue**.

Trois types de diagrammes: diagrammes **fonctionnels**, diagrammes **statiques** et diagrammes **dynamiques**

❖ Vue Statique :

- ✓ **Diagramme de classes** : donne une description statique du système, intégrant une partie relative aux données et une partie relative aux traitements.
- ✓ **Diagramme d'objets** : comporte des instances des classes.
- ✓ **Diagramme de composants** décrit les différents constituants logiciels d'un système.
- ✓ **Diagramme de déploiement** : décrit l'architecture technique d'un système, en présentant :

Eléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage, etc.)
Manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux.

❖ Vue Fonctionnelle :

✓ **Diagramme des cas d'utilisation**

Représente les fonctionnalités que doit fournir le système ;

Décrit les possibilités d'interaction entre le système et les acteurs.

✓ **Diagramme d'activités**, donne une vision de l'enchaînement des activités propres à une opération ou à un cas d'utilisation.

✓

❖ **Vue Dynamique :**

✓ **Diagramme d'états-transitions** : montre la manière dont l'état d'un objet est modifié en réaction aux événements.

✓ **Diagramme de séquence** : décrit les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

✓ **Diagramme de collaboration** : décrit les scénarios des cas d'utilisation présentés par l'intermédiaire d'objets et messages échangés.

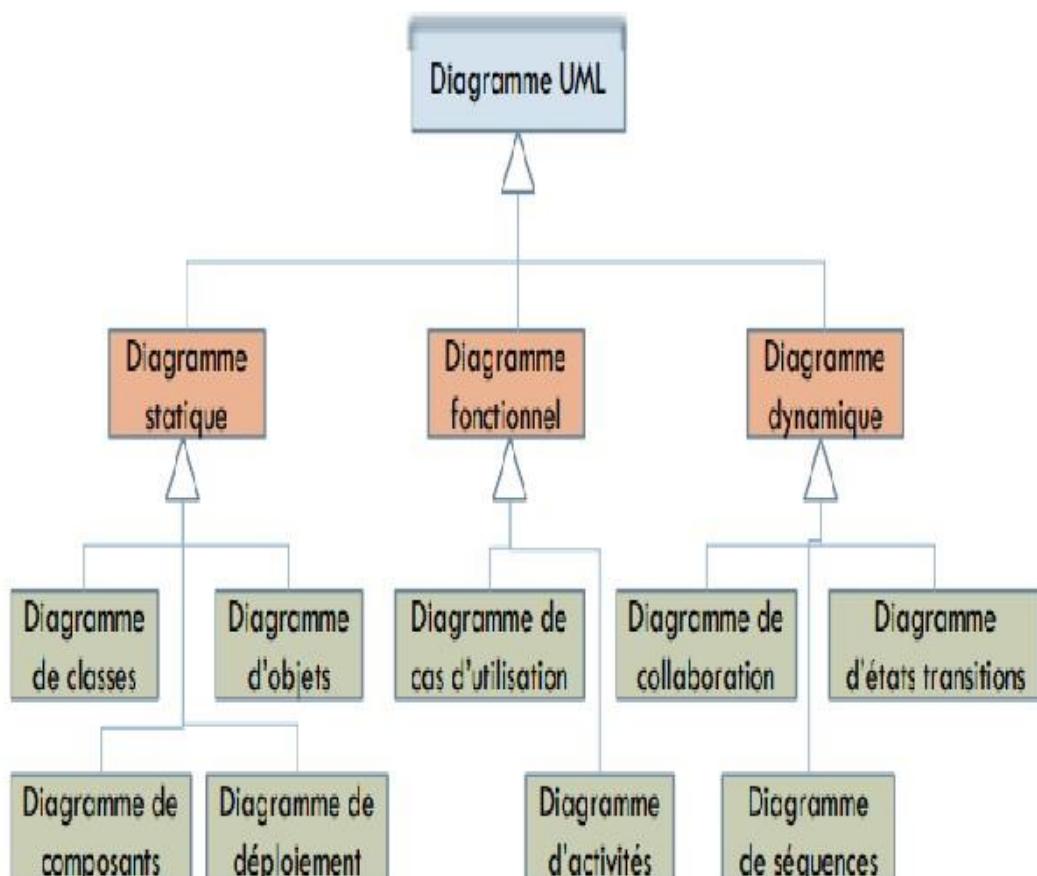


Figure 3: Diagrammes UML 1.4

III.1.2 Version 2.5 de UML

Cette nouvelle version d'UML compte aujourd'hui quatorze (14) diagrammes répartis en trois types de diagramme qui sont :

❖ Diagramme de Structure ou Diagrammes Statiques

- Diagramme de classes ;
- Diagramme de structure composite ;
- Diagramme d'objets ;
- Diagramme de composants ;
- Diagramme de paquetages ;
- Diagramme de déploiement ;
- Diagramme de profils ;

❖ Diagramme de comportement

- ✓ Diagramme de cas d'utilisation ;
- ✓ Diagramme d'états-transitions ;
- ✓ Diagramme d'activité ;

❖ Diagramme d'interaction constitué de :

- ✓ Diagramme de communication ;
- ✓ Diagramme global d'interaction ;
- ✓ Diagramme de séquence ;
- ✓ Diagramme de temps.

Diagrammes UML

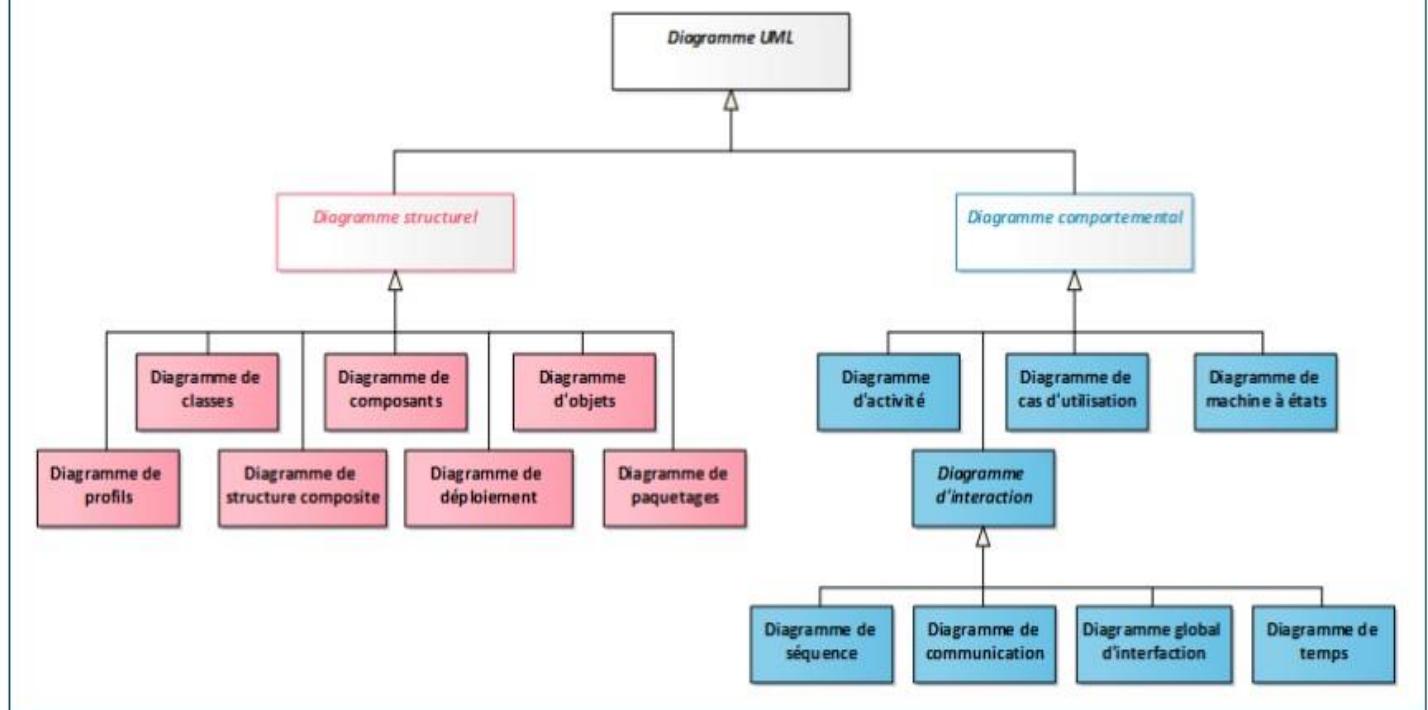


Figure 4:Organigramme UML version 2.5.X

IV. Langage de programmation

Un langage de programmation est un langage informatique, permettant à un être humain d'écrire un code source qui sera analysé par une machine, généralement un ordinateur.

Il existe plusieurs types de langage de programmation : les langages de programmation orienté objet, les langages de programmation procédurale...

IV.1 Langage de programmation de procédurale

La programmation procédurale est un paradigme de programmation basé sur le concept d'appel procédural. Une procédure, aussi appelée routine, sous-routine, méthode ou fonction (à ne pas confondre avec les fonctions de la programmation fonctionnelle reposant sur des fonctions mathématiques) contient simplement une série d'étapes à réaliser. N'importe quelle procédure peut être appelée à n'importe quelle étape de l'exécution du programme, incluant d'autres procédures voire la procédure elle-même (récursivité).

Voici quelques caractéristiques des langages de programmation procéduraux :

1. **Programmation impérative** : Ils suivent un style de programmation impératif, où les instructions sont données dans un ordre séquentiel pour effectuer des actions sur des données.
2. **Procédures** : Les programmes sont organisés en procédures, également appelées fonctions ou sous-programmes, qui regroupent des instructions liées pour accomplir une tâche spécifique.
3. **Variables** : Ils utilisent des variables pour stocker et manipuler des données. Ces variables peuvent être globales ou locales aux procédures.
4. **Contrôle de flux** : Ils offrent des structures de contrôle de flux telles que les boucles (for, while) et les instructions conditionnelles (if, else) pour contrôler l'exécution du programme.
5. **Modularité** : Ils encouragent la modularité en permettant la réutilisation de code via des procédures, ce qui facilite la maintenance et la compréhension du code.

Exemples de langages de programmation procéduraux : C, Pascal, Fortran (dans une certaine mesure), BASIC (dans une certaine mesure).

IV.2 Langage de programmation orienté objet

La programmation orientée objet (POO) est un modèle de programmation qui repose sur le concept de classes et d'objets. Elle structure un programme logiciel en éléments de code simples et réutilisables appelés classes, qui servent à créer des instances individuelles d'objets. La POO facilite la modélisation des systèmes complexes en permettant la définition de structures de données personnalisées et leur manipulation à travers des méthodes. Les principaux langages orientés objet incluent Java, Python, PHP, Perl, Ruby, etc.

IV.2.1 Etude comparative

PHP

PHP (Hypertext Preprocessor) est un langage de script côté serveur, conçu principalement pour le développement web. Créé par Rasmus Lerdorf et apparu en 1995, PHP est largement utilisé pour écrire des applications web exécutées sur des serveurs et est indépendant du navigateur.

❖ Les avantages du langage du PHP :

- **Open Source** : PHP est gratuit et bénéficie d'une large communauté qui contribue à son développement et à ses bibliothèques.
- **Indépendant du Système d'Exploitation** : PHP fonctionne sur divers systèmes d'exploitation comme Windows, Linux, et macOS, ce qui le rend flexible pour différents environnements de serveur.
- **Facilité d'Apprentissage** : PHP a une courbe d'apprentissage douce, ce qui le rend accessible aux développeurs débutants.
- **Large Écosystème** : PHP dispose d'un vaste écosystème de frameworks et de bibliothèques (comme Laravel et Symfony) qui facilitent le développement rapide.
- **Intégration** : Il s'intègre bien avec des bases de données telles que MySQL et PostgreSQL et est souvent utilisé avec des serveurs comme Apache et Nginx.

❖ Les inconvénients du PHP

- Les failles de sécurité se sont révélées très nombreuses au fil des années
- Il n'y a pas de structure prédéfinie, il faut donc ajouter des cadres d'applications pour atteindre des niveaux comme le MVC (CodeIgniter, Symfony...).
- Absence de débuggeur

Python

Python est un langage de programmation de haut niveau, interprété et orienté objet. Il a été développé par Guido van Rossum et publié en 1991.

❖ Avantages

- Il est gratuit.
- Il n'a pas beaucoup de faille de sécurité.
- Grande stabilité du code à travers le temps.
- Le langage est littéraire et simple à comprendre.
- Le langage est soutenu par **Google**.

❖ Inconvénients

- Problèmes dus à la gestion automatique des types de données

Perl

Perl est un langage faiblement typé, en d'autre terme une donnée n'aura pas spécialement de type : les nombres, les chaînes de caractères, les booléens ne seront différenciés que par leurs valeurs et par le contexte de leurs utilisations.

❖ Avantages

- la bibliothèque est très riche en possibilités ;
- il est gratuit ;
- il ne nécessite pas beaucoup de code pour obtenir un résultat ;
- il est très puissant pour manipuler de façon brute du code **HTML** ;
- il est très proche des commandes du système d'exploitation : Facilité de développer des projets nécessitant l'utilisation de nombreuses commandes ou d'applications non-WEB.

❖ Inconvénient

- il est compliqué à utiliser pour les débutants ;
- peu de professionnels savent utiliser le plein potentiel de ce langage ;
- peu documenté en français par les sites Web altruistes.

IV.2.2 Choix et justification du langage de programmation

Nous avons choisi le PHP qui est un langage POO pour notre projet de gestion des immobilisations et amortissements.

PHP est un langage de programmation très populaire utilisé pour la conception des applications web. Plus de 20 millions de sites web sont conçus à l'aide de PHP. Il existe de nombreux langages utilisés pour le développement ou la programmation de sites web. Mais parmi tous ces langages, PHP est le langage de script Web le plus populaire. Il est conçu principalement pour le développement web mais est également utilisé comme langage de programmation à usage général.

V. Système Gestion de Bases de Données Relationnelles (SGBDR)

Un système de gestion de base de données est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des données dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Nous pouvons citer à titre d'exemple : MySQL, PostgreSQL, Oracle.

V.1 Etude comparative

MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) qui fonctionne comme un serveur. Vous pouvez stocker des données dans des tables, les interroger avec des requêtes SQL et les utiliser dans des applications Web. MySQL a été créé par sa société mère, Oracle Corporation, et est disponible sous la licence GNU General Public License (GPL).

❖ Avantages de MySQL

- MySQL est un logiciel gratuit et open-source,
- Facilité d'utilisation : MySQL est conçu pour être facile à utiliser,
- Grande performance : MySQL est conçu pour gérer des bases de données de grande taille et est capable de gérer des millions d'enregistrements en un temps record. Il est également optimisé pour gérer des charges de travail élevées et des connexions simultanées.
- Grande communauté : MySQL bénéficie d'une grande communauté de développeurs et d'utilisateurs qui contribuent à son développement et à sa maintenance. Cela signifie qu'il y a une grande quantité de ressources disponibles pour aider les utilisateurs à résoudre les problèmes et à améliorer les fonctionnalités

❖ Inconvénients de MySQL

- **Limitations de mémoire :** MySQL peut avoir des limitations de mémoire lorsqu'il est utilisé pour gérer des bases de données de très grande taille.
- **Manque de fonctionnalités avancées :** MySQL manque de certaines fonctionnalités avancées que l'on trouve dans d'autres systèmes de gestion de bases de données, comme la gestion de transactions distribuées ou la réPLICATION de bases de données.
- **Manque de scalabilité :** MySQL peut être limité dans sa capacité à gérer des charges de travail élevées et des connexions simultanées en utilisant des configurations par défaut.

- **Manque de sécurité** : MySQL ne possède pas certaines fonctionnalités de sécurité avancées qui sont disponibles dans d'autres systèmes de gestion de bases de données, telles que la gestion des accès à base de données et la gestion des priviléges.

PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle orienté objet puissant et open source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes

❖ Avantages de PostgreSQL

- PostgreSQL peut exécuter des sites Web et des applications Web dynamiques en tant qu'option de pile LAMP.
- PostgreSQL le code source est disponible gratuitement sous une licence open source.
- Pour apprendre PostgreSQL, vous n'avez pas besoin de beaucoup de formation car il est facile à utiliser.
- Faible maintenance et administration pour une utilisation intégrée et en entreprise de PostgreSQL.

❖ Inconvénients de PostgreSQL

- De nombreuses applications open source prennent en charge MySQL, mais peuvent ne pas prendre en charge PostgreSQL
- Sur les mesures de performances, c'est plus lent que MySQL.

IV.2 Choix et justification

MySQL est la base de données open source la plus populaire au monde. Bien qu'elle soit avant tout connue pour son utilisation par des sociétés Web, telles que Google, Facebook et Yahoo!,

MySQL est conçu pour gérer des bases de données de grande taille et est capable de gérer des millions d'enregistrements en un temps record. Il est également optimisé pour gérer des charges de travail élevées et des connexions simultanées.

VI. Technologie de développement

Le développement informatique fait référence au processus de création et d'amélioration des logiciels et des systèmes informatiques, en utilisant les langages de programmation et les outils technologiques appropriés.

Cependant on distingue plusieurs technologies de développement dont les technologies Web, mobile et desktop

V.1 Technologie Web

La technologie web, également connue sous le nom de technologie Internet, englobe tous les outils, langages de programmation, protocoles et normes utilisés pour créer et maintenir des applications et des sites web sur Internet. Cela inclut tout, depuis les langages de balisage tels que HTML (HyperText Markup Language) et XML (eXtensible Markup Language), les langages de style comme CSS (Cascading Style Sheets), jusqu'aux langages de programmation côté serveur tels que PHP, Python, Ruby, et JavaScript.

Elles englobent également des frameworks et des bibliothèques populaires du côté du front-end, ainsi que des outils côté serveur. Les bases de données telles que MySQL, Mongo DB et PostgreSQL sont également essentielles pour stocker et récupérer des données dans les applications web

V.1.1 Framework

Un Framework en informatique ou infrastructure logicielle est un ensemble de bibliothèques, d'outils, de conventions de codage, et de modèles de conception qui simplifient et accélèrent le processus de développement logiciel.

- React.js** : Un Framework JavaScript open-source développé par Facebook, largement utilisé pour la construction d'interfaces utilisateur interactives et dynamiques.
- Angular** : Un Framework JavaScript développé par Google, qui est utilisé pour construire des applications web à grande échelle avec une architecture modulaire.
- Vue.js** : Un autre Framework JavaScript open-source, réputé pour sa simplicité et sa flexibilité, souvent utilisé pour construire des applications web interactives et dynamiques.

4. **Django** : Un Framework web Python haut niveau qui encourage un développement rapide et une conception propre et pragmatique. Il est souvent utilisé pour construire des applications web complexes et évolutives.
5. **Laravel** : Un Framework web PHP élégant et expressif, qui vise à rendre le développement web avec PHP plus agréable et plus rapide en fournissant des fonctionnalités utiles.

V.2 Technologie mobile

La technologie mobile fait référence à l'ensemble des outils, des plates-formes et des technologies utilisées pour développer des applications destinées à être utilisées sur des appareils mobiles tels que les smartphones et les tablettes. Cette technologie a connu une croissance exponentielle ces dernières années avec la prolifération des appareils mobiles et l'augmentation de la connectivité Internet.

V.3 Technologie Desktop

La technologie desktop fait référence à l'ensemble des outils, des plates-formes et des technologies utilisées pour développer des logiciels et des applications destinées à être exécutées sur des ordinateurs de bureau et des ordinateurs portables. Ces applications sont conçues pour fonctionner sur des systèmes d'exploitation de bureau tels que Windows, MacOs et Linux.

VII. Architecture logicielle

Une architecture logicielle est une représentation abstraite d'un système exprimée essentiellement à l'aide de composants logiciels en interaction via des connecteurs, elle représente la structure des structures d'un système.

Elle inclut :

- Les composants logiciels.
- Les propriétés externes visibles de ces composants.
- Les relations entre ces composants

VI.1 L'architecture Modèle Vue Contrôleur (MVC)

L'architecture Modèle Vue Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à spécifier trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- **Le modèle** est un objet qui contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité.
- **Vue :** La vue est une Interface Homme Machine. La vue fait l'interface avec l'utilisateur, sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur. Ses différents événements sont envoyés au contrôleur.
- **Contrôleur :** Le contrôleur est un objet qui permet de faire le lien entre la vue et le modèle lorsqu'une interaction utilisateur est survenue sur la vue. C'est cet objet qui aura pour rôle de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue, puis de renvoyer à la vue les données afin d'être affichées.

Avantages :

L'approche MVC apporte de réels avantages:

- ✓ Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
- ✓ Un gain de temps de maintenance et d'évolution du site.
- ✓ Une grande souplesse pour organiser le développement du site ; indépendance des données, de l'affichage et des actions.

Inconvénients :

- ✓ Augmentation de la complexité lors de l'implantation ;
- ✓ Architecture complexe pour des petits projets ;
- ✓ Le nombre important de fichiers représente une charge non négligeable dans un projet

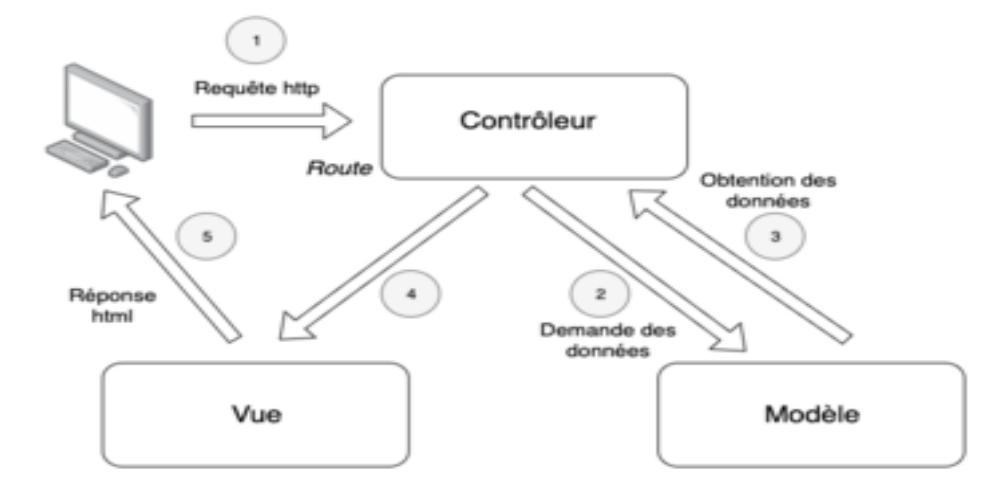


Figure 5 : Figure MVC

VI.2 architecture client/serveur

C'est une architecture informatique visant à répartir une application entre une ou plusieurs unités fonctionnelles qui émettent des requêtes (côté client) et une unité qui traite ces requêtes (côté serveur).

Client : Un client est un système (programme, ordinateur) demandant l'exécution d'une opération à un fournisseur des services par l'envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.

Les caractéristiques d'un client :

- il est actif.
- il émet des requêtes vers le serveur.
- il attend et reçoit les réponses du serveur.

Serveur : Un serveur est un système (programme, ordinateur de grande capacité) détenant des ressources qu'il met à la disposition des autres ordinateurs (clients) d'un réseau. Nous distinguons plusieurs types de serveur en fonction des services rendus : serveur d'application, serveur de base de données, serveur des fichiers.

Les caractéristiques d'un serveur :

- il est passif.
- il est à l'écoute, prêt à répondre aux requêtes envoyées par les clients. Dès qu'une requête lui parvient, il la traite et envoie une réponse au client.

Avantages :

Le modèle client-serveur est particulièrement recommandé pour les réseaux nécessitant un grand niveau de fiabilité, ses principaux avantages sont :

- Une meilleure sécurité : car le nombre de points d'entrées permettant l'accès aux données est moins important.
- Une administration au niveau de serveur : l'administrateur de serveur s'occupe de la gestion de réseau.
- Un réseau évolutif : grâce à cette architecture il est possible de supprimer ou de rajouter des clients sans perturber le fonctionnement du réseau ou procéder à des modifications majeures.

Inconvénients :

L'architecture client-serveur a tout de même quelques lacunes parmi lesquelles :

- Complexité de mise en œuvre : une complexité qui se manifeste notamment lors du découpage des traitements entre partie client et partie serveur.

- Fragilité relative du système : dans un environnement centralisé, les données sont concentrées en un seul endroit, le contexte client/serveur peut nécessiter l'installation des données sur des serveurs distincts. Cette répartition géographique des données, et l'utilisation même d'un réseau fragilise l'ensemble du système et pose alors le problème de la sécurité des données en environnement reparti.

- Investissement initiaux élevés : la mise en place d'une architecture client/serveur nécessite des efforts financiers importants, investissement en matériels et logiciels tels que : micro, interfaces graphiques, matériels de connexion.

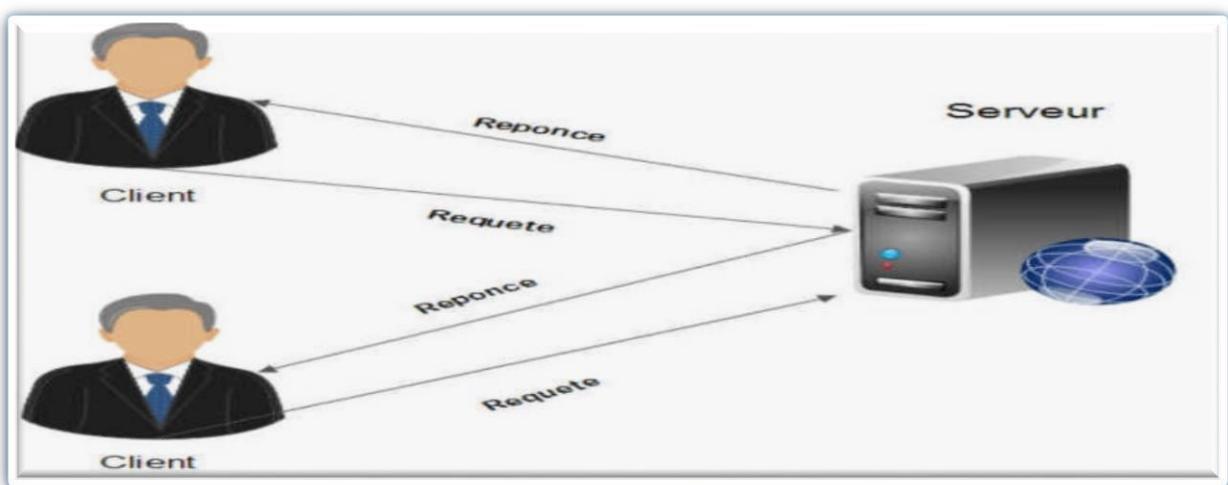


Figure 6 : Figure client /serveur

VI.3 choix de l'architecture

Pour le choix l'architecture de notre système, notre choix se porte sur le modèle MVC tout simplement parce qu'il aide à créer des applications qui séparent les différents aspects de l'application (logique d'entrée, logique métier et logique de l'interface utilisateur) en assurant un couplage lâche entre ces éléments. Le modèle spécifie l'emplacement où chaque genre de logique doit figurer dans l'application.

TROISIEME PARTIE : CADRE ANALYTIQUE

CHAPITRE I : ANALYSE ET CONCEPTION

I. ANALYSE

Notre travail dans cette partie est de présenter l'étude de l'existant du système actuel au sein Gamma Informatique, proposer une solution et présenter l'étude des besoins.

I.1 Etude de l'existant

I.1.1 Analyse de l'existant

L'analyse de l'existant a pour objectif de comprendre la configuration actuelle des systèmes en place et d'identifier les points forts et les faiblesses. Actuellement, Gamma Informatique utilise une application de comptabilité pour enregistrer les prix des immobilisations lors de leur acquisition. En parallèle, la gestion des immobilisations est effectuée via une fiche Excel qui sert d'inventaire et de suivi. Cette méthode présente plusieurs limitations :

- **Manque d'Intégration** : La gestion des immobilisations n'est pas intégrée dans une application dédiée, ce qui implique des tâches manuelles et des risques d'erreur.
- **Suivi et Gestion** : Le suivi des immobilisations se fait de manière rudimentaire, ce qui complique la gestion et la recherche des informations sur les immobilisations.
- **Absence de Calculs Automatisés** : Il n'y a pas de mécanisme automatisé pour le calcul des amortissements, la gestion des dotations ou des cessions, ce qui nécessite des calculs manuels et augmente le risque d'erreurs.

I. 1.2 Critique de l'existant

Après avoir examiné le système actuel, plusieurs problèmes ont été identifiés :

- **Enregistrement Manuel** : L'enregistrement des immobilisations est fait manuellement à l'aide d'un classeur ou de fichiers Excel. Cette méthode est sujette aux erreurs et peut être chronophage.
- **Gestion Déconnectée** : La gestion des immobilisations est effectuée en parallèle avec une application de comptabilité, ce qui entraîne une absence d'intégration et de cohérence dans les données.
- **Difficultés de Suivi** : Le suivi des immobilisations est difficile, car il repose sur des documents non structurés et des systèmes peu adaptés pour une gestion efficace.

- **Recherche des Immobilisations** : La recherche d'informations spécifiques sur les immobilisations est compliquée en raison de l'absence de fonctionnalités de filtrage et de recherche avancées dans les outils actuels.
- **Absence de Calculs Automatisés** : Le système actuel ne permet pas le calcul automatique des amortissements, ni la gestion des dotations et des cessions, ce qui requiert des interventions manuelles et peut entraîner des erreurs.

I.2 Solution envisagées

Face aux insuffisances du système actuel, nous proposons de concevoir et de réaliser une application web pour améliorer la gestion des immobilisations chez Gamma Informatique. Cette nouvelle solution vise à :

- **Indépendance de l'Application** : Créer une application autonome, non liée à d'autres applications, permettant une gestion centralisée et dédiée des immobilisations.
- **Gestion Efficace des Actifs** : Faciliter la sauvegarde, la recherche, et le suivi des matériels informatiques avec une interface utilisateur intuitive et des fonctionnalités avancées.
- **Calcul des Amortissements** : Automatiser le calcul des amortissements pour une gestion précise du cycle de vie des actifs, réduisant ainsi le risque d'erreurs manuelles.
- **Gestion des Cessions** : Permettre la gestion des cessions d'immobilisations, avec un suivi détaillé des transactions.
- **Gestion des Dotations** : Intégrer la gestion des dotations aux amortissements pour une meilleure visibilité et contrôle des coûts

I.3 Etude des besoins

I.3.1 Identification des acteurs

Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé.

Dans le tableau suivant, nous allons identifier les différents acteurs qui interagiront avec le système ainsi que leurs rôles.

Acteurs	Rôles
L'administrateur	Il a la possibilité de gérer tout le système, c'est-à-dire qu'il a accès à toutes les fonctionnalités et tâches du système. Il gère les utilisateurs et leurs droits d'accès et assure la maintenance du système.
Comptable	Il a droit à quelques tâches sur l'application telles que la recherche, l'enregistrement d'un matériel etc. Ses tâches sont limitées.

Tableau 2:Tableau des acteurs des systèmes

I.3.2 Besoins fonctionnels

Ils permettent d'avoir une vue globale des fonctionnalités du système à mettre en place.

Le tableau représente les fonctionnalités de l'outil de conception à mettre en place dans le cadre de ce projet :

Gestion des authentifications	Cette fonctionnalité permet au système d'identifier et faire authentifier chaque acteur. Il permet de conserver la confidentialité des données et de contrôler l'action de chacun des acteurs.
Gestion des immobilisations	Cette fonctionnalité permet aux utilisateurs de consulter, enregistrer, modifier, supprimer et rechercher un actif
Gestion des amortissements	Cette fonctionnalité permet au système de calculer les amortissements d'un actif au cours de sa durée de vie
Gestion des cessions	Cette fonctionnalité permet de céder une immobilisation au vue d'une panne, ou un remplacement
Gestions des dotations	Cette fonctionnalité permet de calculer l'ensemble des dotations pour une année et les biens qui sont concernés

Tableau 3:Tableau des besoins fonctionnels

I.3.4 Besoins non fonctionnels

Le système doit :

- Avoir une interface conviviale permettant un dialogue simple avec l'utilisateur ;
- Être fiable et sécurisé : il impose l'authentification à chaque utilisateur ;
- Avoir un menu bien organisé ;
- Avoir des informations claires et précises ;
- Être portable : fonctionner sur les différents systèmes et de types de navigateur ;

II. CONCEPTION

Dans la phase de conception, nous apportons plus de détails à la solution et la clarification des aspects techniques, tels que la représentation des différents diagrammes (diagramme de classes, de cas d'utilisation, de séquence, d'activité et de déploiement).

II.1 Diagramme de Cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme qui nous permet de définir toutes les fonctionnalités du système.

Elle décrit également les possibilités d'interactions entre les acteurs et le système.

II.1.1 Les relations entre les cas d'utilisation

La relation d'inclusion : Un cas A est inclus dans un cas B si le comportement décrit par le cas A est inclus dans le comportement du cas B : on dit alors que le cas B dépend de A.

La relation d'extension : Si le comportement de B peut être étendu par le comportement de A, on dit alors que A étend B. Une extension est souvent soumise à condition.

La relation de généralisation : Un cas A est une généralisation d'un cas B si B est un cas particulier de A.

II.1.2 Représentation du diagramme cas d'utilisation

Les diagrammes ci-dessous représentent les cas d'utilisation de l'administrateur et du comptable.

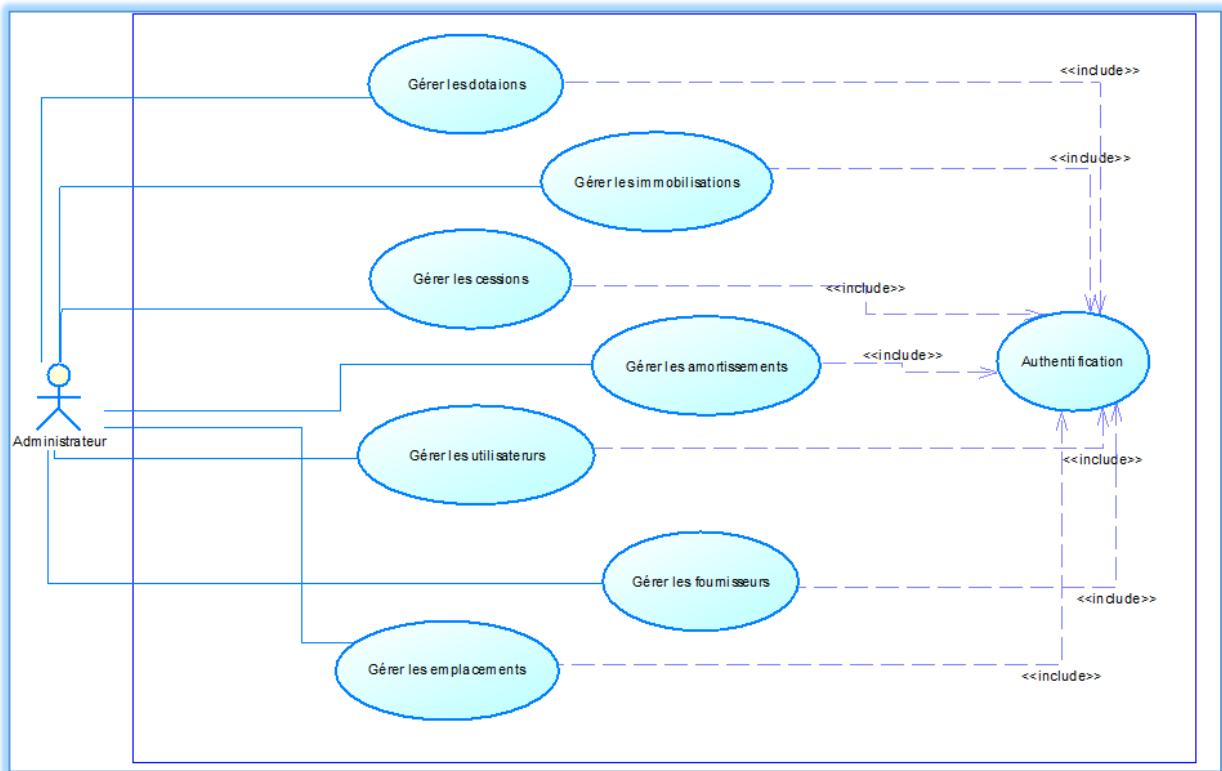


Figure 7:Diagramme de cas d'utilisation pour l'administrateur

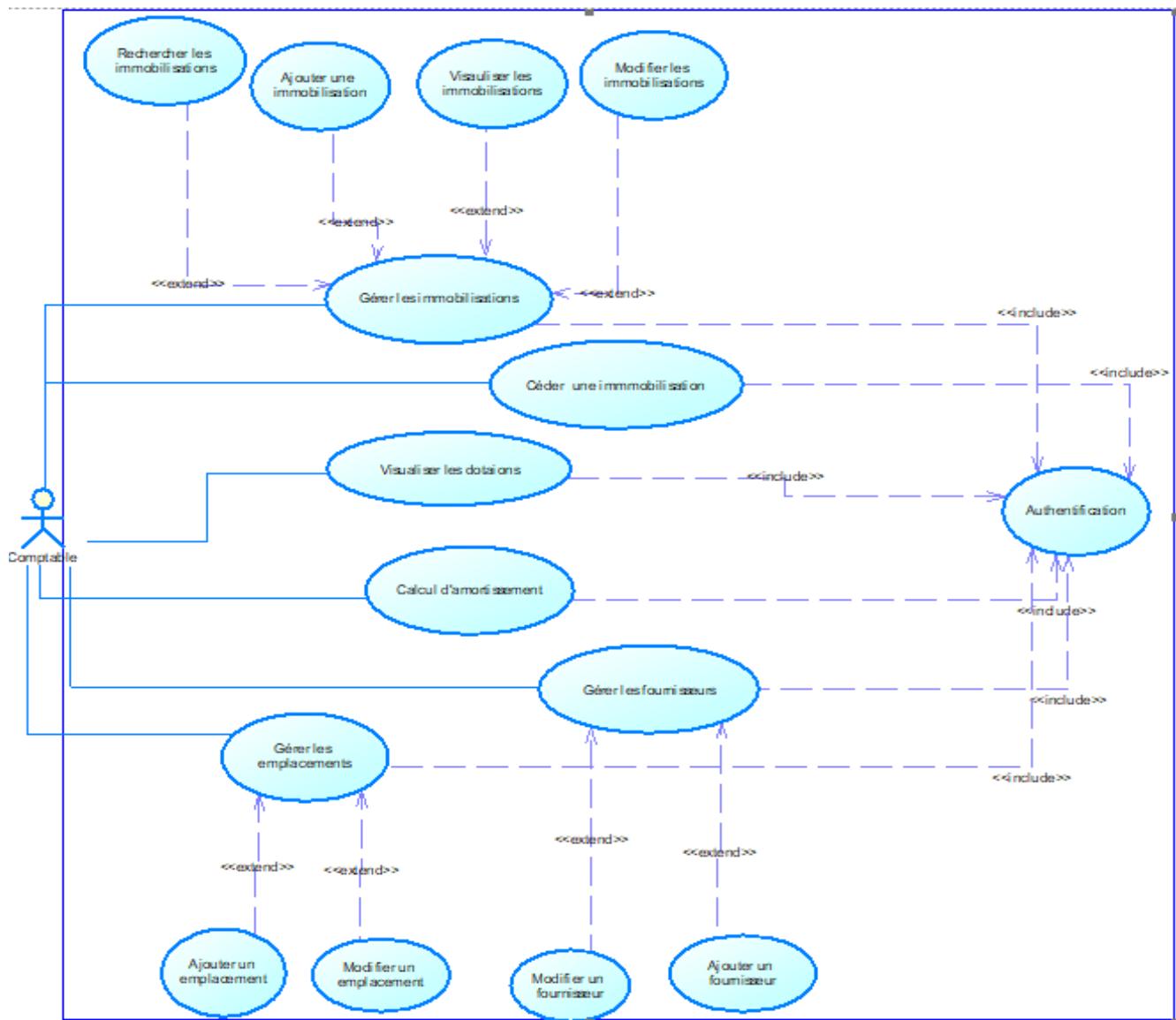


Figure 8 : Diagramme de cas d'utilisation pour le comptable

Ce tableau représente les différents scénarios du cas d'utilisation « s'authentifier » pour pouvoir accéder aux fonctionnalités du système

Description du cas d'utilisation : « S'authentifier »	
Résumé : ce cas d'utilisation permet à l'utilisateur de s'authentifier avant d'accéder au système.	Acteurs : Administrateur, comptable
Scénario nominal	
<p>L'utilisateur fait une demande de formulaire ;</p> <ol style="list-style-type: none"> 1. Le système lui affiche la page de connexion ; 2. L'utilisateur saisit son E-mail et son mot de passe ; 3. Le système vérifie les informations ; 4. Il récupère le profil de l'utilisateur ; 5. L'utilisateur est connecté ; 6. Il peut accéder à la page des fonctionnalités. 	
Scénario d'exception	
<ul style="list-style-type: none"> ❖ le mot de passe ou le login saisi est incorrect. ❖ L'enchaînement peut démarrer après le point 2 de l'enchaînement nominal : ❖ Le système affiche une erreur d'identification. 	

Tableau 4: Description textuelle du cas d'utilisation « authentification »

Ce tableau représente les différents scénarios du cas d'utilisation « Gérer immobilisation » pour pouvoir accéder aux fonctionnalités du système.

Description du cas d'utilisation : « Ajout immobilisation »	
Résumé : ce cas d'utilisation permet à l'utilisateur d'enregistrer un actif	Acteurs : Administrateur, comptable
Précondition	
L'application est démarrée ; L'utilisateur s'est authentifié.	
Scénario nominal	
1. Le comptable clique sur le lien Immobilisation du menu. 2. Il verra le bouton Ajouter et clique. 3. Le système affiche le formulaire d'ajout d'un bien. 4. Le comptable remplit le formulaire. 5. Le comptable clique sur le bouton « Ajouter ». 6. Le système vérifier la validité des données saisies. 7. Le système affiche un message de confirmation de l'opération d'ajout.	
Scénario d'exception	
L'enchainement commence au point 6 du scénario nominal. 6. Le système affiche un message d'erreur. Le système reprend du point 4 de l'enchainement nominal.	

Tableau 5 : Description textuelle « d'ajout immobilisation »

Description du cas d'utilisation : « Calcul d'amortissement »

Résumé : ce cas d'utilisation permet de décrire les étapes permettant à un utilisateur de calculer l'amortissement.	Acteurs : Administrateur, comptable
Précondition	
L'application est démarrée ; L'utilisateur s'est authentifié.	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur sélectionne le bien pour calculer son amortissement. 2. L'utilisateur clique sur le bouton amortissement. 4. Le système calcul l'amortissement et retourne la page d'affichage des informations sur l'amortissement du bien immobilier.

Tableau 6 : Description textuelle de « calcul d'amortissement »

Description du cas d'utilisation : « Ajouter un compte utilisateur »	
Résumé : ce cas d'utilisation permet décrire les étapes permettant à un Administrateur d'ajouter un compte utilisateur.	Acteurs : Administrateur, comptable
Précondition	
<p>L'application est démarrée ;</p> <p>L'administrateur doit s'authentifier.</p>	
Scénario nominal	
<ol style="list-style-type: none"> 1. L'administrateur accède à ajouter compte. 2. L'administrateur peut créer un utilisateur en saisissant et sélectionne les informations 3. L'administrateur clique sur le bouton « Ajouter ». 4. Le système vérifier la validité des données saisies. 	
Scénario d'exception	
<p>L'enchainement commence au point 4 du scenario nominal.</p> <p>6. le système affiche un message d'erreur indiquant la donnée manquante.</p> <p>Le système nominal reprend au point 2.</p>	

Tableau 7 : Description textuelle du cas d'utilisation « ajout compter utilisateur»

II.2 Diagramme d'activités

C'est un diagramme qui représente l'état d'exécution d'un mécanisme sous la forme d'un déroulement d'étapes regroupées séquentiellement.

II.2.1 Représentation du diagramme d'activité de quelque cas

Diagramme d'activités du cas d'utilisation « S'authentifier »

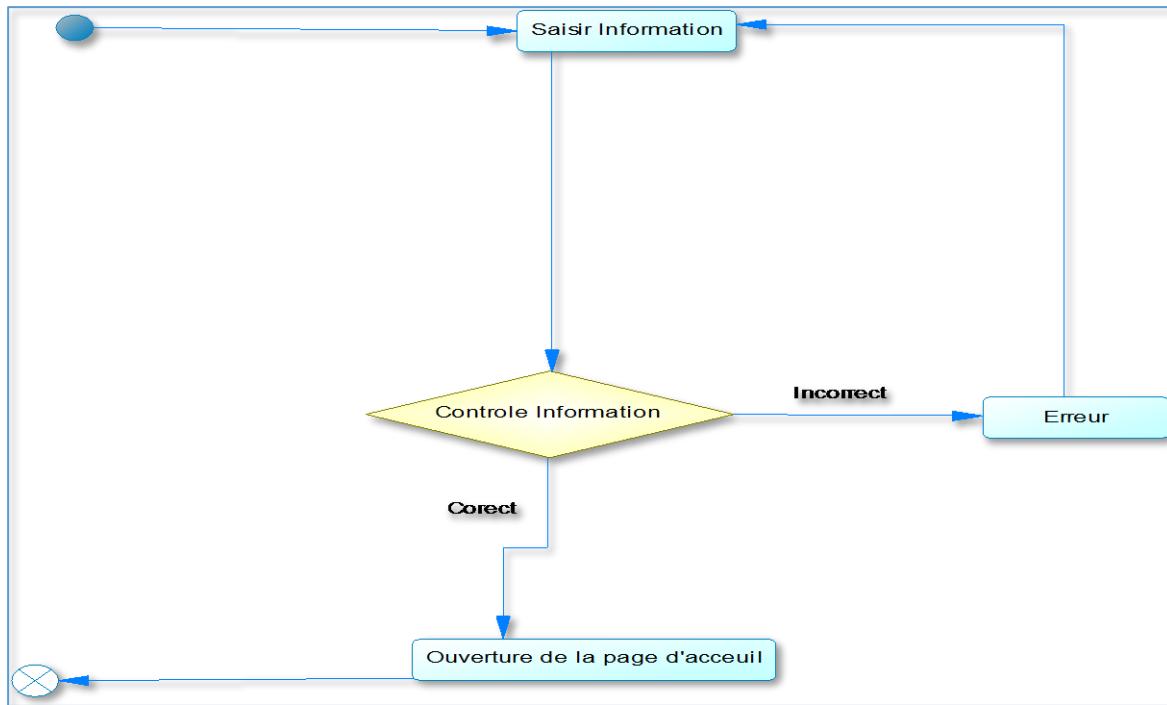


Figure 9 : Diagramme d'activité du cas d'utilisation « Authentification »

Présentation du diagramme d'activité « Ajout immobilisation »

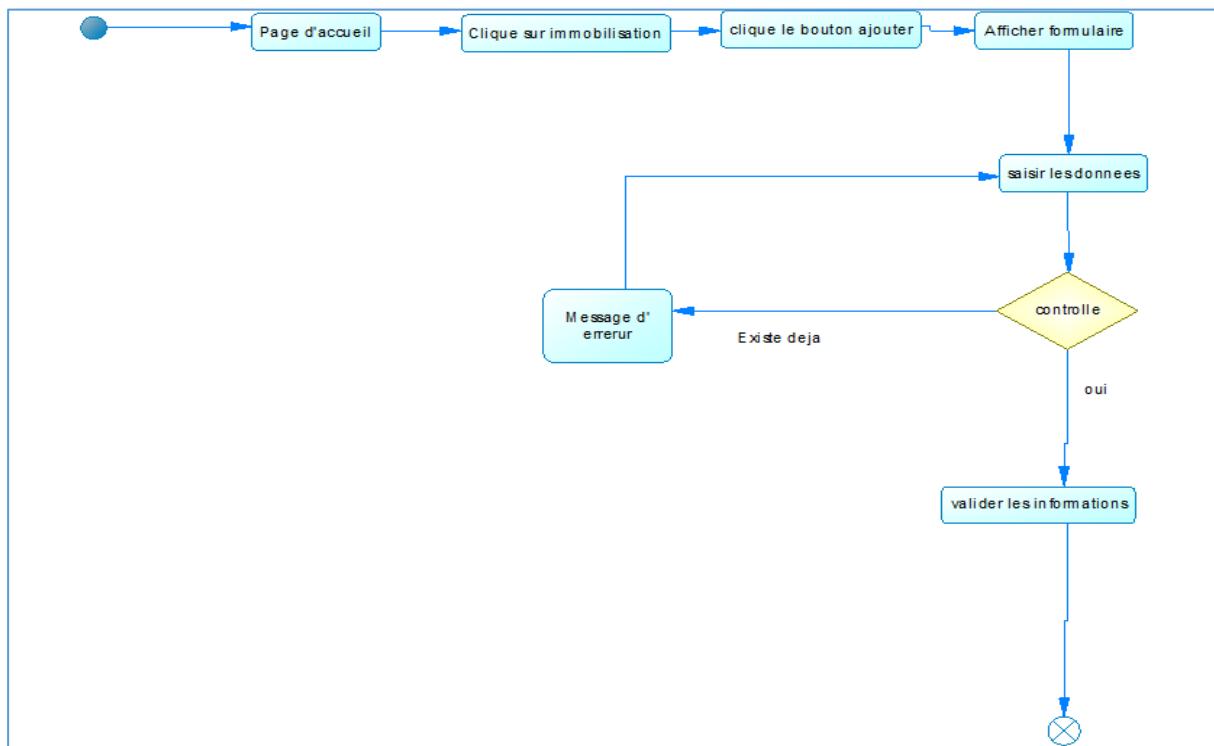


Figure 10 : Diagramme d'activité du cas d'utilisation « ajouter immobilisation »

Nous avons une représentation du diagramme d'activité du cas d'utilisation « modifier immobilisation »

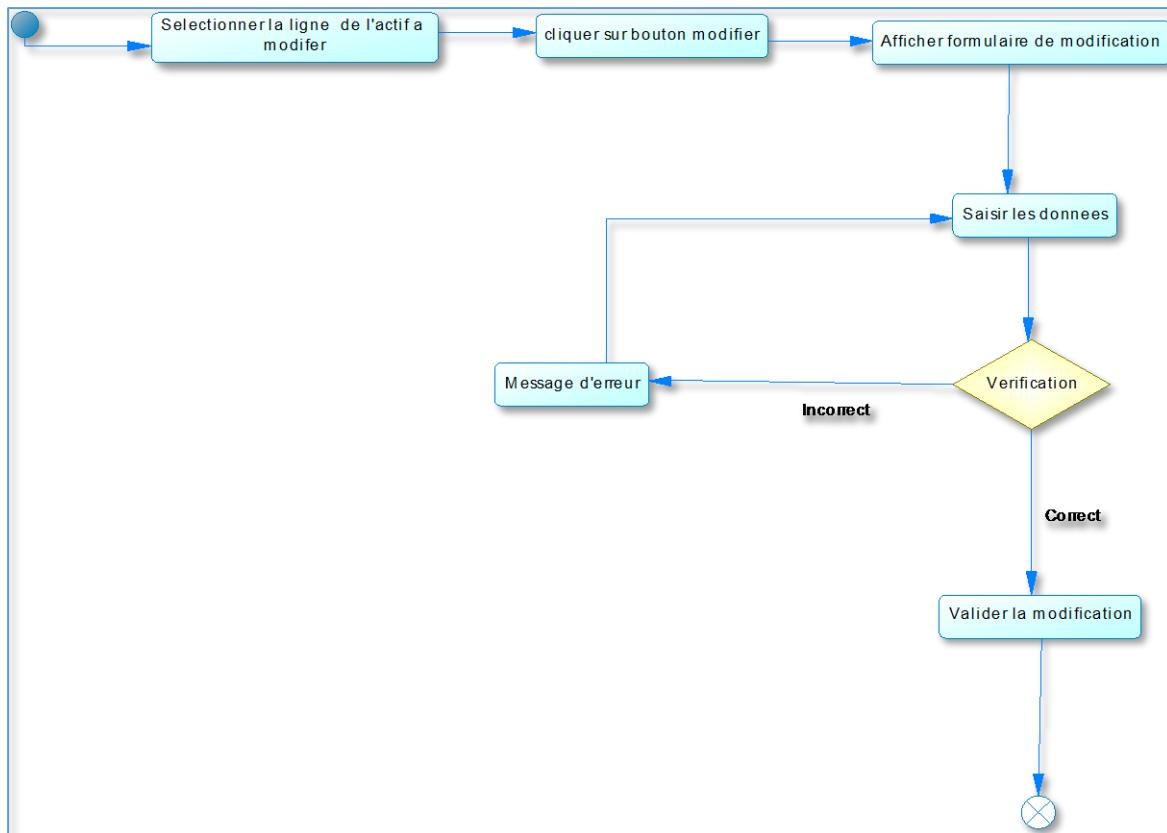


Figure 11 : Diagramme d'activité « modifier immobilisation »

Nous avons une représentation du diagramme d'activité du cas d'utilisation « modifier immobilisation »

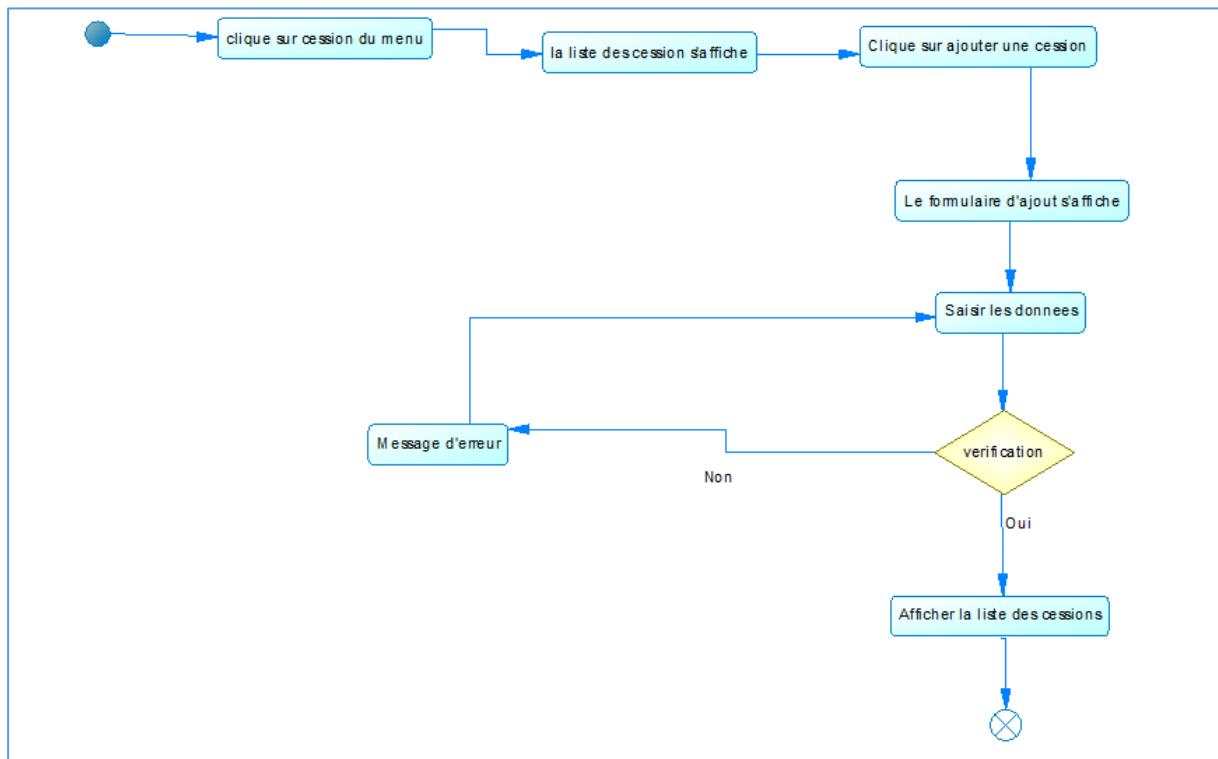


Figure 12 : Diagramme d'activité « ajouter une Cession »

II.3 Diagramme de séquence

C'est un diagramme qui permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec le système.

II.3.1 Représentation de quelques diagrammes de séquence

Nous avons une présentation pour le cas d'utilisation « ajouter immobilisation »

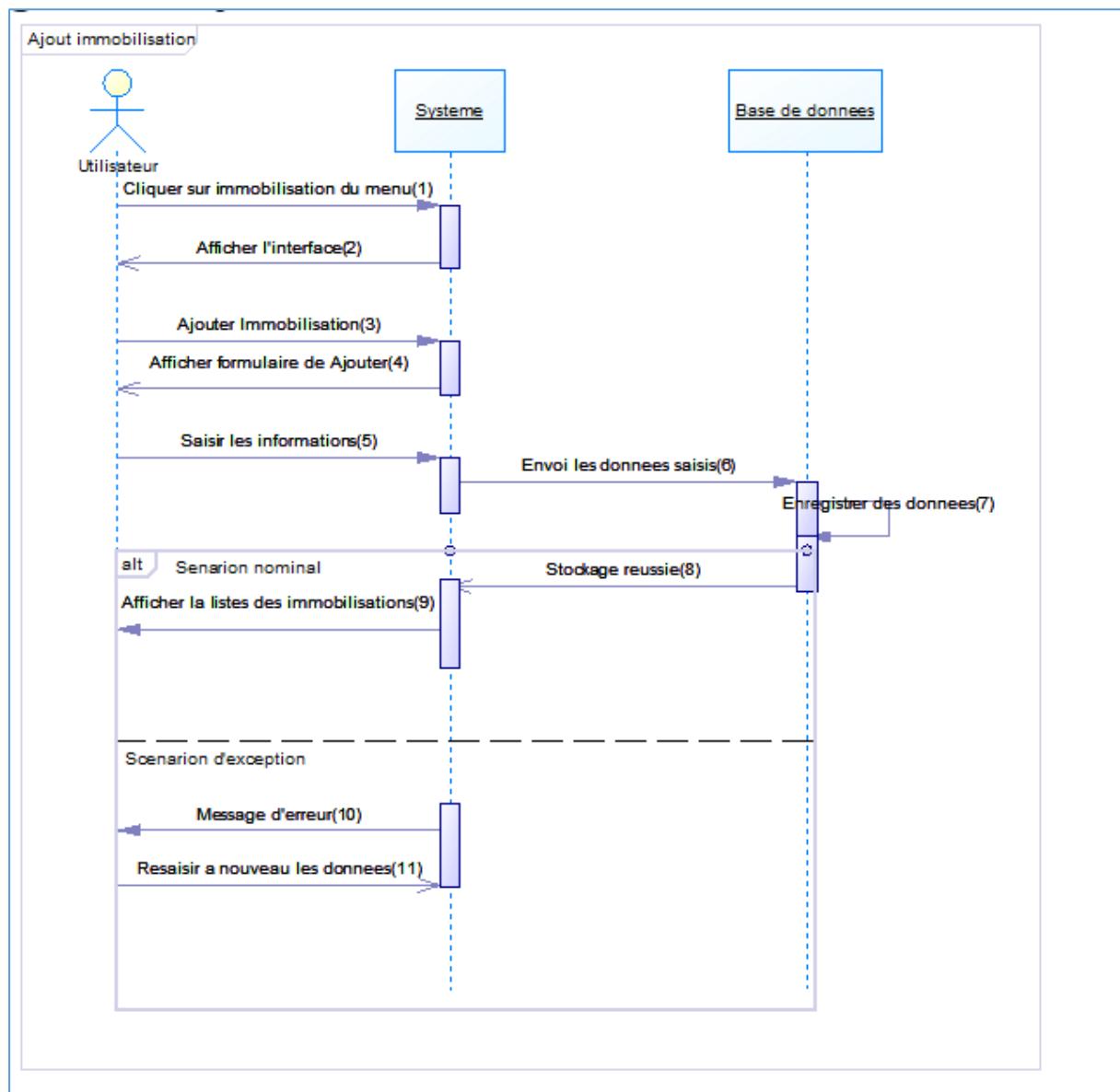


Figure 13 : Diagramme de séquence du cas d'utilisation « Ajouter immobilisation »

Diagramme de séquence pour le cas d'utilisation « calcul amortissement » :

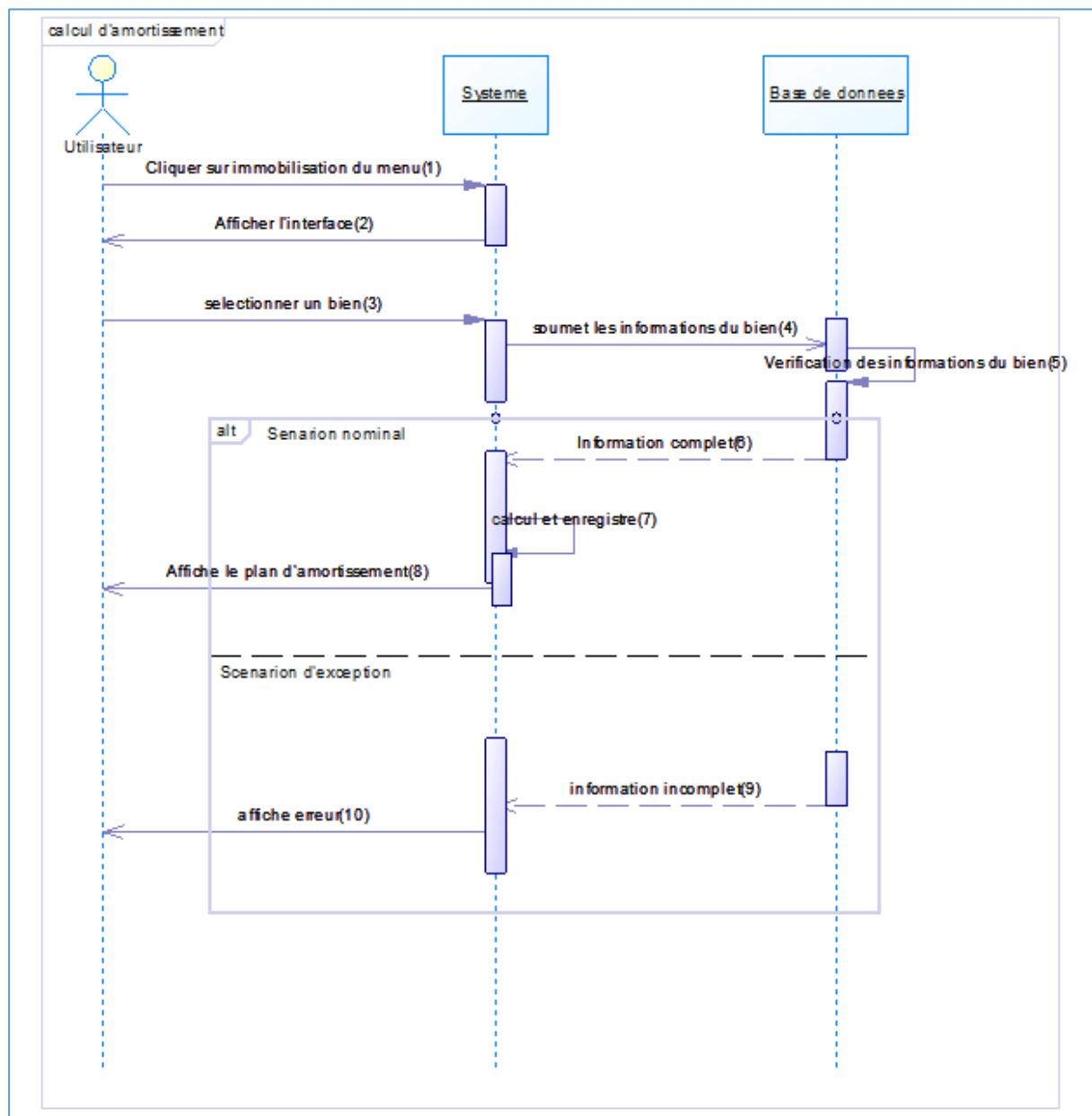


Figure 14 : Diagramme de séquence pour le cas d'utilisation « calcul amortissement »

Diagramme de séquence le cas d'utilisation « céder immobilisation »

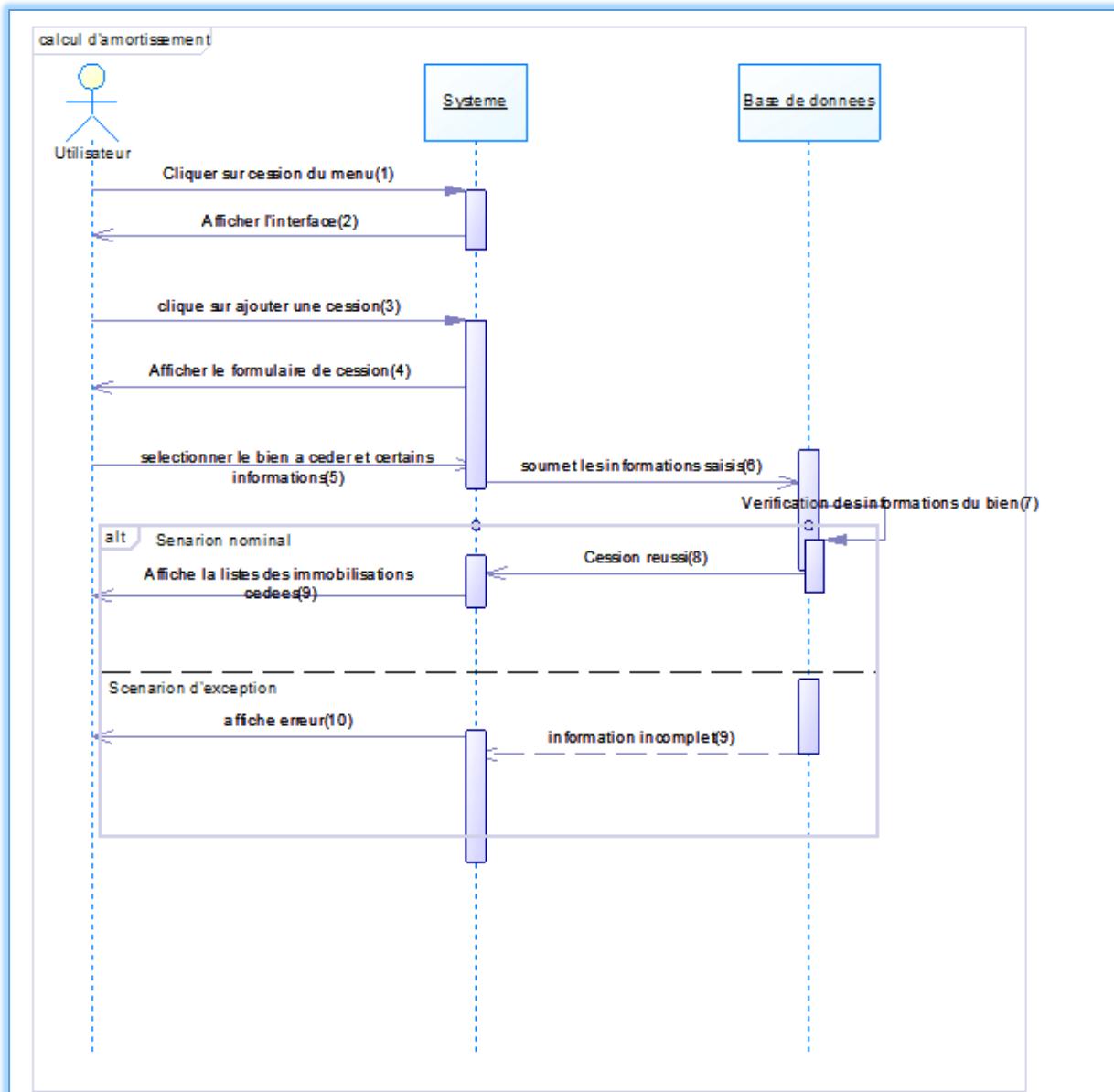


Figure 15 : Diagramme de séquence « céder immobilisation »

II.4 Diagramme de classe

Les diagrammes de classes permettent de représenter une description statique du système.

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

II.4.1 Représentation du diagramme de classe

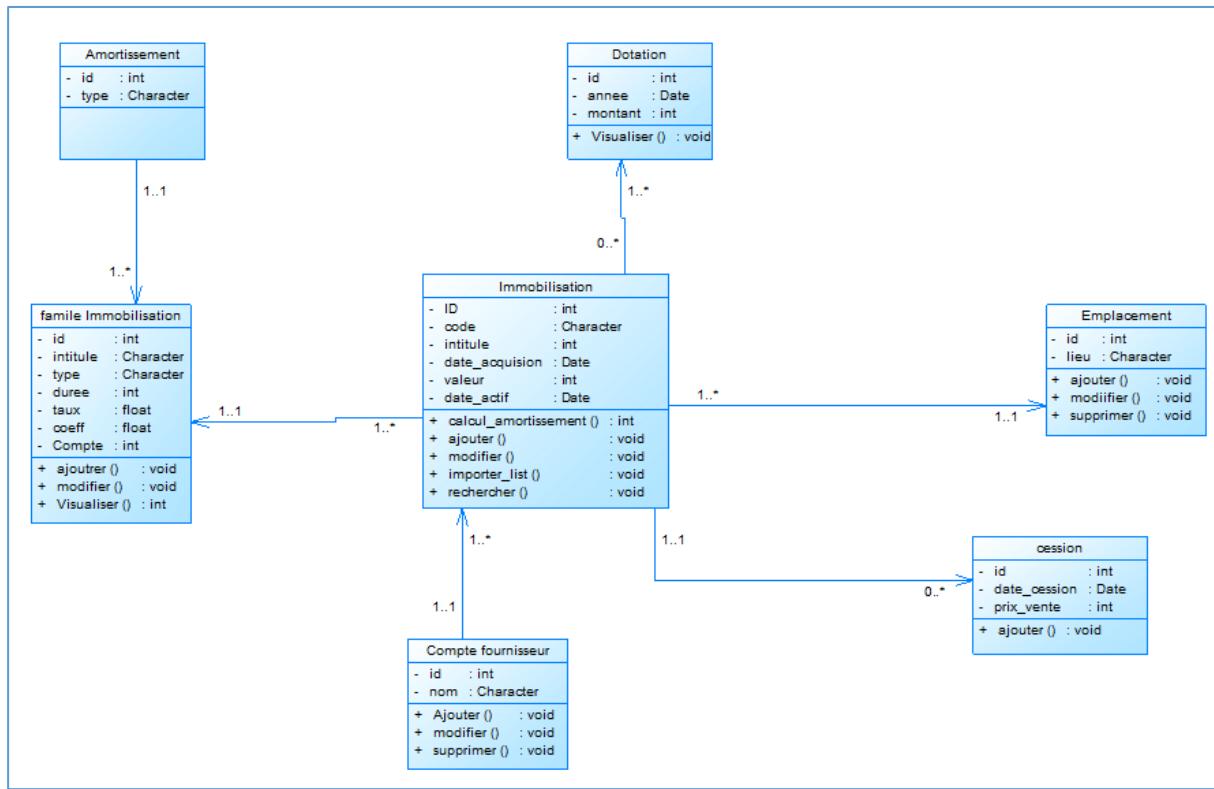


Figure 16 : Diagramme globale su système

II.5 Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

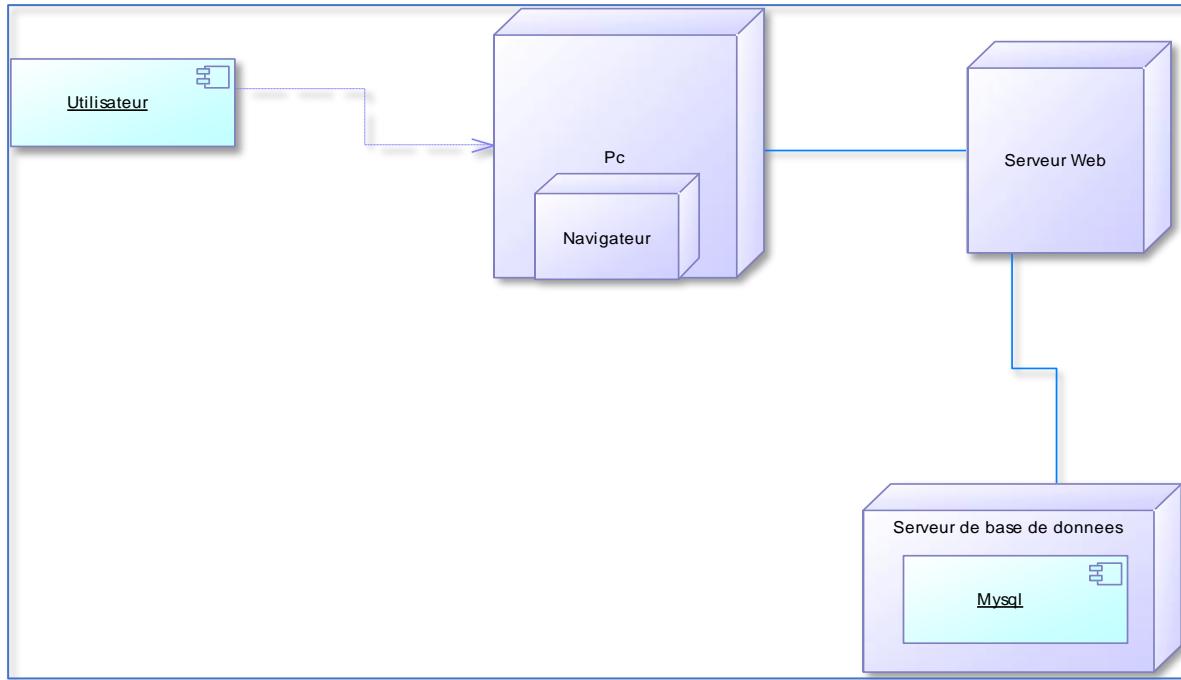


Figure 17 : Diagramme de déploiement

Synthèse

Ce chapitre nous a permis de présenter le volet analyse et conception de notre travail.

Dans la partie « Analyse » au terme de l'étude de l'existant de définir les besoins de notre système.

Dans la partie conception de représenter les différents diagrammes de classes, cas d'utilisation, d'activité, de séquence et de déploiement et enfin choisir l'architecture de notre système.

Le prochain chapitre sera consacré à la partie réalisation.

CHAPITRES II : REALISATION

Après avoir finalisé la phase d'analyse et de conception, nous passons à la phase de réalisation de notre projet informatique.

Ce chapitre est destiné à la présentation des outils et langages de programmation que nous avons utilisés lors de la réalisation de ce projet, de la politique de confidentialité et présentés quelques interfaces de notre application.

I. Outils et librairies utilisés

I.1 Outils de modélisation

i. Power AMC

PowerAMC est un logiciel de conception créé par la société SDP, qui permet de modéliser les traitements informatiques et leurs bases de données associées. Crée par SDP sous le nom AMCDesignor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner. PowerAMC permet de réaliser tous les types de modèles informatiques.



I.2 Outils de développement

i. Visual Studio Code

VS code est un éditeur de code source extensible développé par Microsoft pour Windows, Linux et MacOs. Il est utilisé par des nombreux développeurs pour écrire et modifier du code dans des différents langages de programmation tels que JavaScript, Python, C++, etc... VS code offre de nombreuses fonctionnalités utiles pour les développeurs, les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, la refactorisation du code et Git intégré. L'interface utilisateurs de Visual Studio Code est simple et intuitive, ce qui fait un choix populaire pour les développeurs.



ii. HTML et CSS

HTML (pour HyperText Markup Language, qu'on peut traduire en « langage de balisage hypertexte ») est le langage utilisé pour structurer une page web et son contenu. Son but est de décrire et structurer des informations (textes, images, médias...) dans le but de les faire afficher dans un navigateur.

Cascading Style Sheets (CSS) est un langage de programmation qui vous permet de déterminer le design des documents électroniques. À l'aide de simples instructions, présentées dans des codes sources clairs, les éléments de la page Web comme la mise en page, la couleur et la police peuvent ainsi être modulés à souhait.

iii. PHP

Pour la réalisation de nos travaux, nous avons choisi PHP comme langage  développement. En effet, Il est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. L'un des avantages majeurs de PHP est sa portabilité. En effet, un script PHP codé sous Windows peut être utilisé sous un environnement Unix sans aucune modification. Tout ceci confirme la puissance, la fiabilité et les performances de PHP qui s'est imposé au vu du nombre de sites l'utilisant.

iv. WampServer

WampServer est une plate-forme de développement Web sous Windows pour des  applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin permettant de gérer plus facilement les bases de données.

v. MySQL



MySQL est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels.

I.3 Librairies

i. Bootstrap

C'est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble de composants qui contient des codes HTML et CSS, des formulaires, des



boutons, des outils de navigation et d'autres éléments interactifs, ainsi que des extensions JavaScript en option.

ii. JavaScript



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

iii. JQuery

Jquery est une bibliothèque JavaScript gratuite et multiplateforme. Compatible  avec l'ensemble des navigateurs Web (Internet Explorer, Safari, Chrome, Firefox, etc.), elle a été conçue et développée pour faciliter l'écriture de scripts.

La bibliothèque jQuery permet, entre autres, de gagner en rapidité dans l'interaction avec le code HTML d'une page Web. Elle figure aujourd'hui parmi les bibliothèques JavaScript les plus utilisées dans le monde pour le développement des sites Internet. Elle ne nécessite aucune installation particulière et peut être téléchargée directement depuis le site officiel de la bibliothèque.

iv. Ajax



L'architecture informatique Ajax (acronyme d'asynchrones JavaScript and XML : JavaScript et XML asynchrones) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005. Ajax combine JavaScript, les requêtes de type XMLHttpRequest, les manipulations du DOM, ainsi qu'un format de données (XML ou JSON), afin d'améliorer maniabilité et confort d'utilisation des applications internet.

II. Politique de confidentialité

La politique de sécurité a pour but de minimiser les risques de dysfonctionnement, d'éviter les incohérences des données, de préserver la confidentialité de la base de données et d'éviter la présence de programmes indésirables sur le réseau. Il s'agit donc

de prendre toutes les dispositions afin de réduire au maximum les effets néfastes des pannes matérielles et logicielles sur le système.

II.1 Sécurisation contre les attaques

Afin de préserver le système contre d'éventuelles catastrophes (incendies, inondations, foudre,), nous préconisons la mise en place d'une stratégie de sauvegarde du système. Dans le choix de cette stratégie nous avons privilégié les techniques de sauvegarde suivantes :

La sauvegarde complète : cette méthode consiste à sauvegarder l'intégralité des données.

La sauvegarde différentielle : seuls les fichiers modifiés depuis la dernière sauvegarde complète sont sauvegardés. Le principal avantage de cette catégorie de sauvegarde est qu'il est entièrement fiable étant donné qu'il ne suffira que de la sauvegarde complète.

II.2 La sécurisation contre les virus

Un virus en informatique est un programme malveillant conçu pour se propager à d'autres, ordinateurs en s'insérant dans les programmes légitimes appelé « hôte ». Il se propage principalement par support de stockage (clés USB, disques durs externes, CD-ROM...) et peut provoquer une déstabilisation du système.

II.3 Confidentialité des données

La confidentialité des données passe par la définition des droits d'accès au système. Définition d'un profil utilisateur au moyen de l'utilisation d'un e-mail et d'un mot de passe, permettant d'offrir à chaque utilisateur les données et le traitement auxquels il a droit.

III. Quelques capture d'écran



Figure 18 : Page d'authentification

Intitulé	Date de Cession	Prix de Vente	Autres Informations	Action
office	2024-07-01	2000.00	hors service	
chaise	2024-07-07	0.00	ras	

Figure 19 : Liste des immobilisations cédées

Ajouter une immobilisation

Famille agencement et installation
Code
Intitulé
Valeur d'acquisition
Date d'acquisition jj/mm/aaaa
Date de mise en service jj/mm/aaaa
Fournisseur

Figure 20 : Formulaire permettant d'ajouter une immobilisation

Détails des dotations

immobilisation	Montant
moto	927.78 fcfa
armoire	4,840.00 fcfa
charrette	12,400.00 fcfa
photoshop	352,083.33 fcfa
voiture	480,277.78 fcfa

Figure 21 : Immobilisations concernées pour une annexe a doté

Conclusion

Ce projet a exploré en profondeur les défis complexes de la gestion des immobilisations et de l'amortissement à travers l'application des méthodes d'analyse comme le Two Track Unified Process (2TUP), qui intègre les principes du Processus Unifié (UP). En débutant par une analyse approfondie de l'existant au sein de Gamma Informatique, nous avons pu définir précisément les besoins fonctionnels et techniques de notre système.

En utilisant le modèle MVC pour structurer notre application, nous avons réalisé des diagrammes détaillés tels que les diagrammes de classes, de cas d'utilisation, d'activité, de séquence et de déploiement. Cette approche méthodologique nous a permis de concevoir une solution robuste et bien structurée, alignée avec les standards de l'ingénierie logicielle moderne.

Ce projet illustre comment l'application rigoureuse des méthodes d'analyse comme l'UP peut transformer les défis complexes en opportunités de développement. En intégrant des technologies telles que PHP et MySQL, nous avons non seulement amélioré la gestion des actifs de Gamma Informatique, mais nous avons également jeté les bases pour une croissance continue et une adaptation future aux besoins du marché. Ce travail représente ainsi une contribution significative à la stratégie informatique de l'entreprise, ouvrant la voie à des innovations futures dans le domaine de la gestion des immobilisations.

Bibliographie

- Cours UML de M. KASSOUM SADE, à L'Ecole Supérieure des Communications Electronique et de la Poste (**ESCEP**), 2023-2024 ;
- Cours Projet Informatique de M. KAILOU DJIBO ISSAKA, à L'Ecole Supérieure des Communications Electronique et de la Poste (**ESCEP**), 2023-2024 ;
- Cours Bases de données de M. ALMOU BACHIROU, à L'Ecole Supérieure des Communications Electronique et de la Poste (**ESCEP**), 2023-2024 ;
- Mlle BELGHADID Malika et Mme OUASSA Amina : Conception et réalisation d'une application web pour la gestion et l'amortissement des immobilisations. Cas: ENIEM de Tizi-Ouzou(Algerie)

Webographie

[https://fr.wikipedia.org/wiki/Processus unifi%C3%A9](https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9) (consulté 10 juillet 2024)

[https://fr.wikipedia.org/wiki/Processus unifi%C3%A9#Comparaison avec d'autres m%C3%A9thodes](https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9#Comparaison_avec_d'autres_m%C3%A9thodes)(consulté le 11 juillet 2024)

<https://www.developpez.net/forums/d649076/general-developpement/alm/methodes/xup/difference-entre-up-merise/>(consulté le 10 juillet 2024)

<https://www.dougs.fr/blog/immobilisations-et-amortissements/>(consulté le 25 juin 2024)

<https://fr.wikipedia.org/wiki/PostgreSQL>(consulté le 11mai 2024)

<https://www.techno-science.net/definition/11446.html>(consulté le 10 mai 2024)

<https://cloud.google.com/learn/what-is-a-relational-database?hl=fr#:~:text=Exemples%20de%20bases%20de%20donn%C3%A9es%20relationnelles,->

[Maintenant%20que%20vous&text=Certains%20des%20syst%C3%A8mes%20de%20gestion,SQL%20Server%20et%20Oracle%20Database.](https://www.techno-science.net/definition/11446.html)(consulté le 10 mai 2024)

<https://kinsta.com/fr/blog/php-vs-python/>(consulté le 3mai 2024)

https://fr.wikibooks.org/wiki/Programmation/Programmation_orient%C3%A9e_objet>Liste_des_langages_orient%C3%A9s_objet(consulté le 25 avril 2024)

<https://www.editions-eni.fr/livre/uml-2-5-initiation-exemples-et-exercices-corriges-5e-edition-9782409024085/la-modelisation-de-la-dynamique>(consulté le 27 avril 2024)

https://www.univ-constantine2.dz/CoursOnLine/Benelhadj-Mohamed/co/grain2_2.html(consulté le 22 mai 2024)