



UNIVERSITE DE FIANARANTSOA

**ECOLE DE MANAGEMENT ET D'INNOVATION
TECHNOLOGIQUE**

**MEMOIRE DE FIN DE CYCLE POUR L'OBTENTION DU DIPLOME DE
LICENCE PROFESSIONNELLE**

MENTION : DEVELOPPEMENT D'APPLICATION ET SYSTEME D'INFORMATION

**MISE EN PLACE D'UNE APPLICATION WEB
POUR L'ADMINISTRATION DE VENTE DE
TICKET AU SERVICE D'ACCEUIL DU PARC
NATIONAL ISALO**

Présenté par :

Monsieur TOKY Anjarampifohazana Harddy

Membres du jury :

- Président : Docteur RASOLOARIJAONA Madison
- Examineur : Madame RABEZANAHARY Hoby
- Rapporteur : Docteur RAKOTONIRAINY Hasina Lalaina
- Encadreur Professionnel : Monsieur RAMBELOMANANA Yves

Année Universitaire 2015-2016

CURRICULUM VITAE

TOKY Anjarampifohazana Harddy

Né le 09 Mai 1991 à Ranotsara-Nord

Célibataire

Email : tokyharddy@outlook.com

N° Téléphone : 034 95 916 92



**Etudiant en Troisième Année de Licence en
Développement d'Application et Système d'Information**

FORMATIONS ET DIPLOME :

- **2015-2016:** Troisième Année de Licence Professionnelle à l'Ecole de Management et d'Innovation Technologique (EMIT) de l'Université de Fianarantsoa,
- **Aout 2016 :** Certificat Internationale de Secourisme, agréé par la CROIX ROUGE MALAGASY,
- **Mai 2016 :** Formation Nationale au Secourisme organisée par la Croix Rouge Malagasy,
- **2014-2015 :** Deuxièmes années de Licence Professionnel au Centre Universitaire de Formation Professionnalisante (CUFP) de l'Université de Fianarantsoa(UF),
- **Juillet 2015 :** Formation du geste de Premier Secours organisée par la CROIX ROUGE MALAGASY Haute Matsiatra,
- **Mai-Aout 2014 :** Formation de « LEADERSHIP », « Wilderness First Aid et CPR and AED », organisée par l'Emergency Care & Safety Institute (ECSEI), à SKY RANCH Lutheran Camp, Fort Collins Colorado Etats-Unis d'Amérique,
- **2012-2013 :** Première Année de Licence Professionnelle en Informatique de Système et Réseaux au Centre Universitaire de Formation Professionnalisante (CUFP) de l'Université de Fianarantsoa(UF),
- **2012 :** Baccalauréat de l'Enseignement General Série C, Lycée Morarano Ihosy.

EXPERIENCES PROFESSIONNELLES :

– **Décembre 2016- Mars 2017 :** Stage professionnel au sein du Madagascar National Parks Parc National Isalo, durée de trois mois, a pour thème « Mise en place d’une application web pour l’administration de vente de l’écotourisme aux Service d’Accueil », développé avec le Framework CodeIgniter PHP avec JetBrains PhpStorm.

- Stage deux (02) mois au service d'accueil du Madagascar National Parks Isalo,
- Secourisme Jeune Volontaire du CROIX ROUGE MALAGASY Fianarantsoa.
- Conception et réalisation d’un logiciel pour la « Gestion de réparation Scooteur » en JAVA, JSP, PHP, C++ et Python.
- Conception et réalisation d’un logiciel pour la « Gestion de note du CUFP » en Access.

CONNAISSANCES EN INFORMATIQUE :

- Systèmes d’Exploitation : Windows, Linux.
- Méthode et langage de conception : MERISE 2, UML.
- Outils de conception : Microsoft Access, Visual Paradigm, Win ‘Design.
- Outils de développement : NetBeans, Eclipse, PhpStorm.
- Programmation Web : HTML, PHP, JSP.
- Langages de Programmation : C, C++, Java, Visual Basic, Pascal.
- SGBD: MS Access, MySQL, PostgreSQL.

CONNAISSANCES LINGUISTIQUES :

Langues	Aptitude à :			
	Comprendre	Lire	Ecrire et Rédiger	Parler et Communiquer Oralement
Malagasy	Très Bien	Très Bien	Très Bien	Très Bien
Française	Bien	Bien	Assez-Bien	Assez-Bien
Anglaise	Bien	Très Bien	Bien	Bien

DIVERS :

- Sport: Basket-ball, Rugby, Volley-ball, Natation.
- Sens du travail d’équipe, sens de la communication.
- Lire des livres, regarder des documentaires et films.

AVANT-PROPOS

Le terme principal de la formation universitaire est de préparer les étudiants dans le domaine du travail. Ce futur métier qui attend les étudiants requiert le savoir-faire, la bonne volonté, le dynamisme, et le professionnalisme. Le futur cadre doit, donc, être à la portée de ces exigences. L'adaptation des méthodes avec l'évolution de la technologie dans le monde devient de plus en plus nécessaire de nos jours. Avec les possibilités qui accompagnent cet essor inévitable, nous pouvons envisager l'informatisation de tous systèmes de travail.

Au sein de l'Université de Fianarantsoa, plus exactement à l'Ecole de Management et d'Innovation Technologie, ses étudiants sont formés pour être des licenciés directement actifs dans le domaine du travail et répondent aux besoins du métier de nos jours. Pour y parvenir, pour bien garantir la formation, et dans son cursus normal des études, les étudiants doivent obligatoirement réaliser un stage dans une Entreprise ou Administration pour perfectionner leur formation théorique apprise et s'améliorer dans le domaine professionnel qui veut dire d'intégrer les étudiants dans le monde de professionnalisme et de les familiariser avec les établissements de travail en faisant une application sur terrain dans une Société.

Etant un de ces étudiants, de la mention Développement d'Application et Système d'Information (DASI), l'obtention du diplôme de licence est donc conditionnée par un stage en entreprise d'une durée de quatre mois au maximum. Alors que c'est durant ce stage que ce mémoire de fin d'étude est rédigé.

Ce mémoire met en avant l'importance et la contribution du web dans nos échanges quotidiens. Notre travail s'insère alors dans l'étude et la mise en place d'une application web. Nous allons donc énoncer dans ce mémoire toutes les démarches que nous avons effectuées lors de cette éducation.

REMERCIEMENTS

Il nous est particulièrement agréable avant de présenter notre travail, d'exprimer toute notre gratitude envers les personnes qui de près ou de loin nous ont apporté leur sollicitude.

Ce présent ouvrage a été le fruit d'une collaboration de différents intervenants à qui nous tenions présenté tous nos sincères remerciements. Plus particulièrement à;

- Professeur RASOAZANANIERA Marie Monique, Ministre de l'Enseignement Supérieur et de la Recherche Scientifique ;
- Docteur RAFAMANTANANTSOA Fontaine, Président de l'Université de Fianarantsoa ;
- Docteur HAJALALAINA Aimé Richard, Directeur de l'Ecole de Management et d'Innovation Technologique - Université de Fianarantsoa (EMIT-UF), de m'avoir accueilli dans son Etablissement ;
- Madame RABEZANAHARY Hoby, Responsable de Mention du Développement d'Application et Système d'Information à l'EMIT-UF ;
- Docteur RAKOTONIRAINY Hasina Lalaina, notre encadreur pédagogique, malgré ses combines fonctions tout en nous encadrant au cours de l'arrangement de cet ouvrage ;
- Monsieur RAKOTOARIVELO Nandrasana Jean Jacques, Directeur du Parc National Isalo (PNI), de m'avoir accordé de faire mon stage dans son Unité de Gestion ;
- Monsieur RAMBELOMANANA Yves, Chef de Volet Écotourisme du Parc National Isalo, notre encadreur professionnel qui nous a conduits dans le milieu sérieux ;
- A tous les enseignants et les personnels de l'EMIT-UF et du PNI ;
- A tous les membres de jury.

Ces remerciements vont également à nos très chères et adorables familles qui sont toujours été présents, avec leurs soutiens et leurs précieuses aides, et à tous ceux qui sont concourus pour accumuler cet ouvrage.

Merci à vous tous !

LISTE DES ABREVIATIONS

AJAX	Asynchronous JavaScript And XML
AED	Automated External Defibrillator
AES	Administration Economique et Sociale
ANGAP	Association National pour la Gestion des Aires Protégées
AP	Agent du Parc
APD	Avant Produit Dérivée
ASP	Active Server Pages
BD	Base de données
CLP	Commutés Locales du Parc
COSAP	Commuté d’Orientation et de Soutien aux Aires Protégées
CFC	Centre de Formation Continue
CI	CodeIgniter
CPR	Cardio-Pulmonary Resuscitation,
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
CU	Cas d’Utilisation
CUFP	Centre Universitaire de Formation Professionnalisante
CUG	Chef de l’Unité de Gestion
CVAF	Chef de Volet Administratif et Financier
DAF	Direction d’Administration Financière
DASI	Développement d’Application et Système d’Information
DEAP	Droit d’Entrée aux Aires Protégés
DED	Demande d’Engagement des Dépenses
EMIT	Ecole de Management et d’Innovation Technologique
Fac DEGS	Faculté de Droit, Economie Gestion et Sciences Social
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IHM	Interface Home Machine
JSP	Java Server Pages
LCD	Liquid Cristal Display
LED	Light Emitting Diode

MD	Management Décisionnel
MERISE	Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise
MESupRE	Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
MNP	Madagascar National Parks
MVC	Modèle Vue Contrôleur
PAF	Plan d'Aménagement de Gestion
PCA	Pièce Comptable d'Accueil
PHP	Personal Home Pages
PNI	Parc National Isalo
POO	Programme Orienté Objet
PTA	Plan de Travail Annuel
PTBA	Plan de Travail Budget Annuel
RAM	Random Access Memory
RPM	Relation Publique et Multimédia
SGBD	Système de Gestion de Base de Données
SIGD	Système d'Information Géomatique et Décisionnel
SQL	Structured Query Language
UF	Université de Fianarantsoa
UML	Unified Modeling Language
URL	Uniform Resource Locator
VP	Visual Paradigme
WAMP	Windows, Apache, MySQL et PHP à l'origine puis « Perl » ou « Python »
W3C	World Wide Web Consortium

LISTE DES FIGURES

Figure 1.1 :	Organigramme de l'EMIT.....	3
Figure 2.1 :	Carte de localisation du Parc National Isalo.....	7
Figure 2.2 :	Organigramme du MNP PNI.....	13
Figure 4.1 :	Modèle de Cycle de Vie d'une application Web.....	21
Figure 4.2 :	Modèle en Cascade.....	24
Figure 4.3 :	Modèle en V.....	25
Figure 4.4 :	Modèle en Spirale	26
Figure 4.5 :	Modèle par incrément ou itératif.....	27
Figure 4.6 :	Modèle incrémentale	28
Figure 4.7 :	Evolution de l'UML	31
Figure 4.8 :	Les différentes diagrammes de l'UML	36
Figure 4.9 :	Formalisme de diagramme de cas d'utilisation.....	37
Figure 4.10 :	Formalisme de diagramme de séquence	39
Figure 4.11 :	Formalisme de diagramme de classe	41
Figure 4.12 :	Formalisme de diagramme d'activité	42
Figure 4.13 :	Formalisme de diagramme de déploiement	42
Figure 6.1 :	Diagramme de cas d'utilisation de l'application	53
Figure 6.2 :	Diagramme de séquence de l'authentification	55
Figure 6.3 :	Diagramme de séquence du cas voir statistique	56
Figure 6.4 :	Diagramme de séquence de cas d'utilisation « Calcul des données»	56
Figure 6.5 :	Diagramme d'activité de l'authentification.....	57
Figure 6.6 :	Diagramme d'activité global de l'application	58
Figure 6.7 :	Diagramme de Classe de l'application	59
Figure 6.8 :	Phase de Conception de l'IHM	60
Figure 6.9 :	Représentation de la structure de la page de l'application	61
Figure 7.1 :	Structure de la base de données	63
Figure 7.2 :	Courbe des recherches relatives au Framework PHP	70
Figure 7.3 :	Architecture MVC	71
Figure 7.4 :	Template CodeIgniter au format PHP	73
Figure 7.5 :	Modèle et Contrôleur	74
Figure 7.6 :	Fonctionnement de CodeIgniter	75
Figure 7.7 :	Fonctionnement d'AJAX	77

Figure 8.1 :	Evolution du web	79
Figure 8.2 :	Architecture d'une application web	82
Figure 8.3 :	Architecture matérielle	83
Figure 9.1 :	Page d'authentification de cette application	86
Figure 9.2 :	Page d'accueil de l'application	86
Figure 9.3 :	Interface de choix du nom pour une facture	87
Figure 9.4 :	Forme de la facture à imprimer	88
Figure 9.5 :	Forme à imprimer de la PCA	88
Figure A.1 :	Répertoire d'application	93
Figure A.2 :	Page d'accueil de l'application de CodeIgniter	95

LISTE DES TABLEAUX

Tableau 3.1 :	Déroulement du stage	16
Tableau 4.1 :	Avantages et Inconvénients de l'UML	34
Tableau 4.2 :	Classement des diagrammes	36
Tableau 4.3 :	Comparaison des outils de conception UML	43
Tableau 5.1 :	Liste des matérielles informatiques et leurs caractéristiques.....	45
Tableau 5.2 :	Liste des logicielles	45
Tableau 5.3 :	Planning du projet	49
Tableau 5.4 :	Liste des équipes du projet et leurs actions	49
Tableau 6.1 :	Dictionnaire de données	52
Tableau 6.2 :	Raffinement du cas d'utilisation s'authentifier	54
Tableau 6.3 :	Description d'une classe	58
Tableau 7.1 :	Comparaison de la base de données	64
Tableau 7.2 :	Comparaison des langages de développement web	68
Tableau 7.3 :	Comparaison des Framework PHP	72
Tableau 8.1 :	Configuration minimale des matérielles et logicielle	81

GLOSSAIRE

Article : Le plan du Parc, en générale, est présenté dans ce plan et un petit souvenir de diverses espèces dans le parc.

Client-Serveur : L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs clients du serveur : chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur est généralement capable de servir plusieurs clients simultanément. Le client et le serveur doivent bien sûr utiliser le même protocole de communication.

Framework : Un Framework est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

IHM : Interface Homme Machine.

Livre de compte : C'est la version physique en papier de la base de données de la vente tous les jours par mois.

Marchandise : C'est la vente des tickets, des Tee-shirt, des Stylo qui ont le logo du MNP.

MVC : Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle.

Note de Débit : C'est la facture pour client avec quoi tous les renseignements utiles pendant la visite sont mentionnés : le type de ticket, le nom de guide, le prix unitaire de ticket, le montant total en nombre et en lettre, la date et la durée de la visite, signature de client et le responsable d'accueil...

Parc : C'est le lieu où le guide amène les visiteurs avec les tickets et la Note de Débit.

PHP : Personal Home Pages est un langage de programmation pour permettre la création de page HTML de manière dynamique.

Routards : Ce sont des livres qui sont écrit pour indiquer les prix, les chemins et toutes possibilités de séjours pendant la visite des clients à Madagascar.

Serveur SGBD : Serveur SGBD Système de Gestion de Base de Données. Logiciel permettant de stocker les données, de les mettre à jour et de les consulter.

Ticket : C'est le billet que le visiteur achète afin d'avoir l'autorisation d'entrée dans le parc.

Vente : Ici c'est la vente de ticket et les autres marchandises qui sont à vendre à l'accueil du parc.

TABLE DES MATIERES

CURRICULUM VITAE	i
AVANT-PROPOS	iii
REMERCIEMENTS	iv
LISTE DES ABREVIATIONS	v
LISTE DES FIGURES	vii
LISTE DES TABLEAUX	ix
GLOSSAIRE	x
TABLE DES MATIERES	xi
INTRODUCTION	1
PARTIE I	
Chapitre 1: Présentation de l'Ecole de Management et d'Innovation Technologique.....	2
1.1. Historique	2
1.2. Mission de l'Ecole	2
1.3. Organigramme de l'EMIT	3
1.4. Partenaires	4
1.5. Cycle de licence.....	4
Chapitre 2 : Présentation du Madagascar National Parks Parc National Isalo.....	7
2.1 Localisation	7
2.2 Historique	8
2.3 Rôle et mission	8
2.4 Profil des Postes.....	8
2.5 Organigramme du MNP PNI.....	13
Chapitre 3 : Description du projet	14
3.1. Cahier des charges	14
3.2. Formulations	17
3.3 Besoins des utilisateurs.....	17

3.4 Moyens pour la mise en œuvre	19
PARTIE II	
Chapitre 4 : Présentation du modèle de cycle de vie et du langage de modélisation utilisés ..	20
4.1. Cycle de vie d'un logiciel	20
4.1.1. Définition d'un modèle.....	20
4.1.2. Modèle de cycle de vie d'un projet web	20
4.1.3. Modèle de cycle de vie d'un logiciel	22
4.1.4. Les différents modèles de cycle de vie	23
4.1.5. Comparaison de modèle de cycle de vie.....	29
4.1.6. Choix et justification	29
4.2. Méthodologie de conception et technologies utilisées	30
4.3 Langage de modélisation utilisée	30
4.3.1. L'aspect choisi	30
4.3.2. L'UML.....	31
4.3.2.1. <i>Historique</i>	31
4.3.2.2. <i>Définitions</i>	31
4.3.2.3. <i>Les caractéristiques de ses diagrammes</i>	32
4.3.2.4. <i>Ses Objectifs</i>	33
4.3.2.5. <i>UML face à la programmation</i>	33
4.3.2.6. <i>Les points forts et points faibles d'UML</i>	33
Chapitre 5 : Analyse détaillée du projet web	44
5.1. Étude préalable	44
5.2. Etude de l'existant	44
5.2.1 Analyse de l'existant	44
5.2.1.1 <i>Matériel existant</i>	45
5.2.1.2 <i>Logiciel existant</i>	45
5.2.1.3 <i>Mode de gestion de vente</i>	46

5.2.2. Etude de faisabilité	46
5.2.3. Critique de l'existant	47
5.2.3.1 <i>La critique positive</i>	47
5.2.3.2 <i>La critique négatives</i>	47
5.2.4. Solutions proposées	47
5.2.5 Solution retenue	48
5.3 Type de l'application	48
5.3.1. Objectifs généraux de l'application	48
5.3.2. Rédaction du plan projet.....	49
5.3.3. Planning du projet web	49
5.3.4. Constitution de l'équipe.....	49
Chapitre 6 : Conception du projet	51
6.1 Règles de Gestion	51
6.2 Dictionnaire de données	51
6.3. Diagramme de cas d'utilisation	53
6.3.1. Identification des acteurs	53
6.3.2. Identification des cas d'utilisation	53
6.3.3. Spécification détaillé des cas d'utilisation d'un système	54
6.4 Diagrammes de séquences	55
6.4.1 Diagramme de séquence de cas d'utilisation «S'Authentifier»	55
6.4.2 Diagramme de séquence de cas d'utilisation «Voir la statistique des données »	56
6.4.3 Diagramme de séquence de cas d'utilisation «Calcule des données»	56
6.5 Diagramme d'activité	57
6.6 Le diagramme de classe.....	58
6.7. Conception des Interfaces Hommes-Machines	59
6.7.1. Phase de conception.....	59
6.7.1.1. <i>Maquette</i>	59

6.7.1.2. <i>Prototype</i>	60
6.7.1.3. <i>Le Produit finale</i>	60
6.7.2. <i>Analyse préalable</i>	60
6.7.3. <i>Structure ou trame de la page</i>	61
6.7.4 <i>Aspect de l'application</i>	62
PARTIE III	
Chapitre 7 : <i>Présentation des outils de réalisation</i>	63
7.1. <i>Choix d'outils de développement</i>	63
7.1.1. <i>Choix du SGBD</i>	63
7.1.1.1 <i>Généralité</i>	63
7.1.1.2. <i>Etude comparative des différents SGBD</i>	64
7.1.1.3. <i>Présentation du SGBD MySQL</i>	64
7.1.2. <i>Choix de langage de Programmation web</i>	66
7.1.2.1 <i>J.S.P</i>	66
7.1.2.2 <i>ASP</i>	67
7.1.2.3 <i>PHP</i>	67
7.1.3. <i>Comparaison des Framework PHP</i>	69
7.1.3.1 <i>Point Fort de framework PHP</i>	69
7.1.3.2 <i>Les Framework PHP</i>	69
7.1.3.3 <i>Les différents frameworks existants</i>	70
7.1.3.4 <i>Choix de Framework</i>	72
7.1.4 <i>Présentation du Framework CodeIgniter</i>	72
7.1.4.1 <i>Définition</i>	72
7.1.4.2 <i>Installation</i>	73
7.1.4.3 <i>Organisation des répertoires</i>	73
7.1.4.4 <i>Template</i>	73
7.1.4.5 <i>Accès aux données</i>	74

7.1.4.6 Cache	74
7.1.4.7 Documentation	74
7.1.4.8 Maturité.....	74
7.2 Présentation du langage et quelques outils de technologie complémentaire.....	76
7.2.1 Le langage HTML	76
7.2.2 Le langage CSS	76
7.2.3 Le langage JAVASCRIPT	76
7.3 Outils de Technologie complémentaire (AJAX/JQuery)	76
7.4. Présentation des outils de développement	77
Chapitre 8 : Mise en œuvre du projet.....	79
8.1. La normalisation	79
8.1.2.1. Les perspectives de conception d'une IHM	80
8.1.2.2. Les risques d'une mauvaise interface	80
8.2. Configuration minimale matérielle et logicielle	81
8.3. Architecture Web.....	82
Chapitre 9 : Présentation de l'application	85
9.1. Fonctionnalités de l'application.....	85
9.2. Présentation de quelques pages de l'application	85
9.2.1. Page d'authentification	85
9.2.2. Page d'accueil	86
9.2.3 La Facture et la PCA.....	87
CONCLUSION	89
BIBLIOGRAPHIE	90
ANNEXE	93

INTRODUCTION

Le parc national Isalo est l'un de deux parcs exceptionnels à Madagascar, il attire beaucoup plus de nombre de visiteurs que les autres. Tous les jours, surtout pendant la haute saison, les hôtes à l'accueil risquent d'en avoir oublié certaines données car leurs travaux sont en explosion en faisant un aller-retour physiquement et numériquement l'enregistrement des données et de le gérer en même temps. Plus particulièrement, le calcul de la recette journalière et celle de la note de débit pendant la vente qui doit être fait pour valider le ticket d'entrée ont besoins d'une amélioration plus vif et avec une certitude.

Il ne fait désormais plus aucun doute que l'informatique est la révolution la plus importante et la plus innovante marquant la vie de l'humanité moderne. Ses logiciels et ses méthodes de conception et de développement ont vu l'avènement de technologies quotidiennement utilisées. Pour cela, est-il possible d'avoir une application qui peut administrer la vente plus active et que le calcul total de la recette sur la note de débit se fait automatiquement, et qui peut aussi gérer leur temps de travail? Ce présent ouvrage constitue le fruit du travail que nous avons accompli dans le cadre de ce stage de fin de cycle. L'objectif est donc « la conception et la réalisation d'une application web pour l'administration de vente de l'écotourisme du Parc National Isalo ».

Durant la période de stage, la première phase était l'étude de l'existant, en établissant un cahier de charges qui définit les besoins réels de l'application, puis chercher une solution optimale qui va respecter tout ce qui a été demandé, ensuite l'élaboration de la phase de mise en œuvre de l'application.

Après certaines recherches préliminaires et grâce à des informations collectées nous avons élaboré notre vision de l'application, son architecture, ainsi que les outils et les technologies utiles. Alors pour arriver à ce but, les différentes étapes de la conception et la réalisation du projet durant le stage sont présentés par ce présent mémoire.

Nous verrons au cours de cet ouvrage, en tripartites, comment administrer les vente au service d'accueil pour répondre à ces problématiques. Dans un premier temps, nous répertorierons comme première partie les présentations générales de l'EMIT et celle du MNP Parc National Isalo. Ensuite, en deuxième partie, l'analyse et conception du projet pour déployer la présentation du projet et la méthode utilisée et le langage de modélisation, l'analyse du projet et en fin la conception. Et pour terminer, nous aborderons comme troisième partie la réalisation en expliquant la présentation des outils de réalisation, la mise en œuvre et la présentation de l'application.

PARTIE I – PRESENTATION GENERALE

Chapitre 1: Présentation de l'Ecole de Management et d'Innovation Technologique

1.1. Historique

L'Ecole de Management et d'Innovation Technologique (EMIT) a été modifiée au sein de l'Université de Fianarantsoa par le conseil du Gouvernement du Mardi 15 novembre 2016 par adoption de textes règlementaires au titre du ministère de l'enseignement supérieur et de la recherche scientifique suivant le Décret N° 2010-1394, il a été connu sous le nom Centre Universitaire de Formation Professionnalisante (CUFP) par le Décret N° 2005/205 du 26 Avril 2005 et prépare le diplôme de Licence Professionnelle en Administration et en Informatique. Auparavant, il a été connu sous le nom de Centre de Formation Continue (CFC), créé par l'arrêté rectoral N°99-23/UF/R du 10 Mars 1999 et dispensait des formations des Techniciens Supérieurs. (BACC+2). L'année Universitaire 2013 - 2014, l'EMIT est basculé totalement vers le système Licence, Master et Doctorat (LMD). En effet, les offres formations de l'école sont habilitées :

- Mention : Administration Economique et Sociale (AES), Grade de Licence (Arrêté N° 8007/2014-MESupRES) ;
- Mention : Développement d'Application et Système d'Information (DASI), Grade de Licence (Arrêté N° 8007/2014-MESupRES) ;
- Mention : Relation Publique et Multimédia (RPM), Grade de Licence (Arrêté N° 8007/2014-MESupRES) ;
- Mention : Management Décisionnel (MD), Grade de Master (Arrêté N° 8007/2014-MESupRES) ;
- Mention : Système d'Information - Géomatique et Décision (SIGD), Grade de Master (Arrêté N° 23411/2014-MESupRES).

1.2. Mission de l'Ecole

L'Ecole a pour mission, d'abord de dispenser des formations initiales et continues en informatique, en administration et en relation publiques. Ensuite, il offre des services connexes à l'informatique. Former des techniciens supérieurs spécialisés, opérationnels immédiatement dans les entreprises. Elle assure le perfectionnement professionnel des étudiants, des demandeurs d'emplois, des employés et des cadres d'entreprises.

1.3. Organigramme de l'EMIT

L'organigramme de l'Ecole de Management et d'Innovation Technologique montre des conseils (établissement, scientifique), une direction, un collège des enseignants, des chefs de mention et services existants dans l'Ecole, il s'est présenté par la figure 1.1.

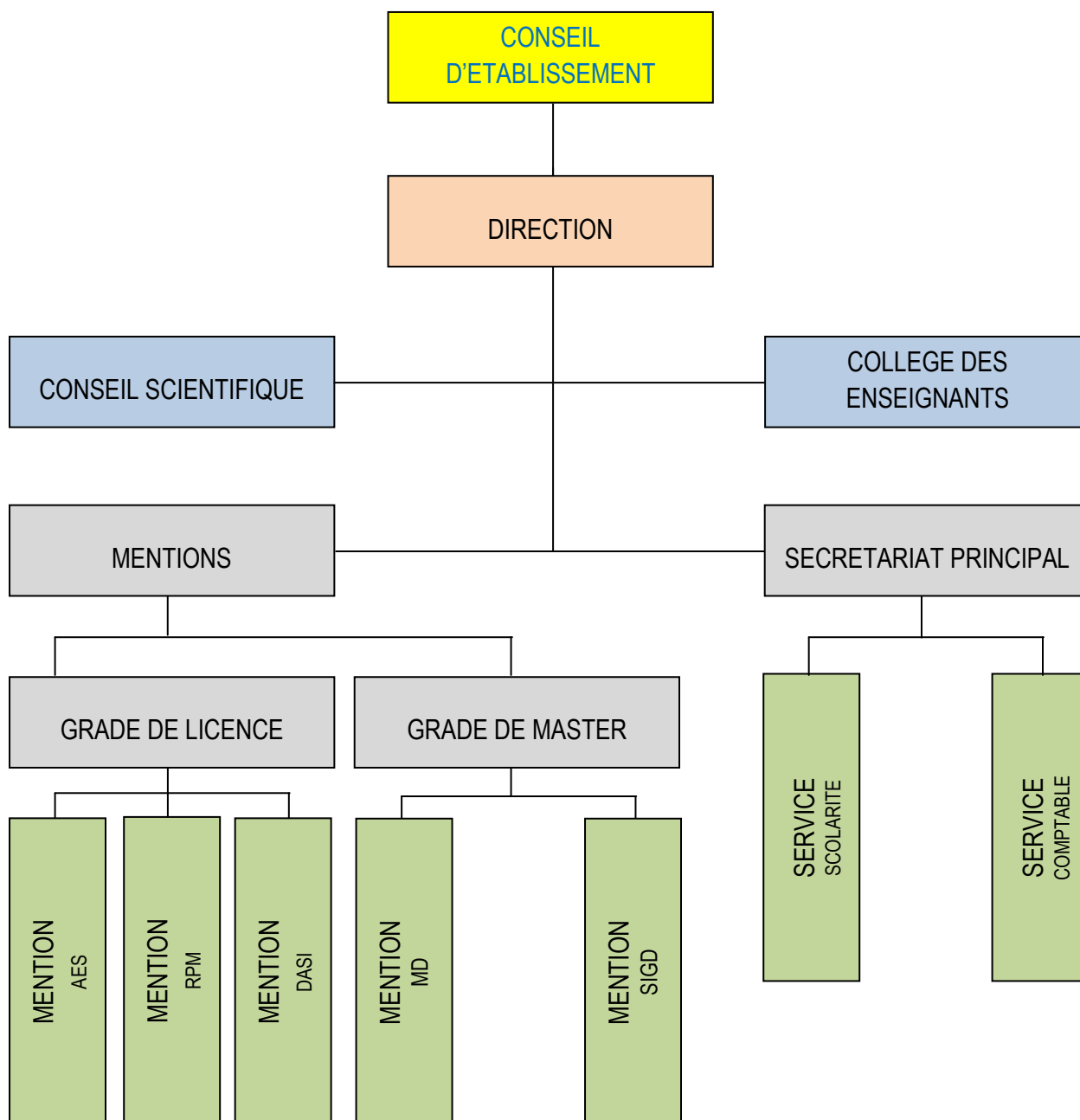


Figure 1.1: Organigramme de l'EMIT

1.4. Partenaires

Le Département des Sciences Sociales (Faculté de DEGS-UF), Le Département d'Economie et de Gestion (Fac DEGS/UF), le Département de Mathématique (Faculté des SCIENCES-UF) et le Département de Droit (Fac DEGS/UF). L'Institut d'Observation Géophysicien Applique (IOGA). Le Foibe Taon-tsaritan'iMadagasikara (FTM). Le ministère des Finances et des Budgets, le ministère de l'Education Nationale, le ministère de Transport et de Météorologie, le ministère de Tourisme. Le BOA, le BFV-SG et le BNI. L'assurance ARO, NY HAVANA et MAMA. La société Lazan'i Betsileo, la société STAR, la société JIRAMA et les réseaux TIAVO sont les partenaires de l'EMIT.

1.5. Cycle de licence

L'EMIT-UF possède trois (03) cycles de Licence Professionnelle dont :

- La Licence Professionnelle en Administration Economique et Sociale (AES);
- La Licence Professionnelle en Développement d'Application et Système d'Information (DASI);
- La Licence Professionnelle en Relations Publiques et Multimédia (RPM).

1.5.1 Condition Générale

Dans la formation de Licence Professionnelle, les conditions sont tous valable pour les trois cycles en licence:

- La condition d'admission : Test de niveau ;
- La condition d'accès : Baccalauréat général de toute série (A, C, D), Baccalauréat technique G1 et G2 pour la filière AES et RPM mais seulement les baccalauréats scientifique pour la filière DASI;
- La durée de la formation : 03 années universitaires.
- Un stage d'insertion en entreprise ou voyage études en première année;
- Un stage de réalisation suivi d'un rapport de stage en deuxième année;
- Un stage de fin de cycle suivi d'une soutenance d'un mémoire en troisième année.

1.5.2. Licence professionnelle en Développement d'Application et Système d'Information (DASI)

Après avoir complété la formation, les étudiants obtiennent un diplôme de Licence en Développement d'Application et Systèmes d'Information et ont les capacités en :

- Administration des bases de données ;

- Administration des réseaux et systèmes informatiques ;
- Développement d'application client/serveur ;

1.5.3. Licence professionnelle en Administration Economique et Sociale (AES)

Comme dans la filière DASI, les étudiants obtiennent un diplôme de Licence en Administration Economique et Sociale.

A l'issue de la Connaissance, les étudiants ont les compétences :

- Assister le Directeur Général, le Directeur des Ressources Humaines (DRH) et le Directeur Administratif et Financier (DAF).
- Gérer des Ressources Humaines.
- Gérer une Entreprise et un projet.

1.5.4. Licence professionnelle en Relations Publiques et Multimédias (RPM)

Les étudiants obtiendront un diplôme de Licence en Relations Publiques et Multimédia, à condition qu'ils accomplissent toute les trois années de la formation, et ont l'aptitude à :

- Rédiger un article dans un journal.
- Occuper une poste d'un technicien de presse.
- Travailler dans la revue de presse.

1.6. Cycle de master

L'Ecole de Management et d'Innovation Technologique (EMIT) de l'Université de Fianarantsoa possède deux (02) cycle de Master Professionnel dont :

- Le Master Professionnel en Management Décisionnel (MD);
- Le Master Professionnel en Système d'Information – Géomatique et Décision (SIGD).

1.6.1. Master professionnel en Management Décisionnel

1.6.1.1. Condition d'accès

- En S7: Sur sélection de dossier après Licence en Administration Economique et Sociale, en Gestion ou en Économie.
- En S9 : Après validation des crédits déjà obtenus par les responsables.

1.6.1.2. Durée de formation

La durée de formation est quatre semestres (deux années universitaires).

1.6.1.3. Objectifs

Le Master Management décisionnel a pour objectif d'équiper l'apprenant avec les outils de prise de décisions en matière de management et de leur donner les compétences requises dans ce domaine .Comme le management a besoin de se conformer en permanence aux diverses nouvelles exigences du marché, l'enseignement doit alors toujours viser à mettre à jour les connaissances de l'apprenant par la formulation de programmes de cours qui tiennent compte de ces nouveautés. Ainsi l'objectif majeur du parcours est d'avoir des acteurs de haut niveau en management décisionnel. Enfin, les parcours préparent des cadres capables de gérer et créer un projet de développement économique régional et national.

1.6.2. Master professionnel en Système d'Information Géomatique et Décision (SIGD)**1.6.2.1. Condition d'accès**

- En S7: Sur sélection de dossier après Licence en Informatique, Gestion, Économie, Mathématiques et Physique.
- En S9 : Après validation des crédits déjà obtenus par les responsables.

1.6.2.2. Durée de formation

La durée de formation est quatre semestres (Deux années universitaires).

1.6.2.3. Objectifs

Le parcours Système d'Information, Géomatique et Décision a pour objectif de donner un panorama des recherches actuelles et émergentes dans le domaine des systèmes d'aide à la décision. Les systèmes informatiques et la géomatique connaissent en effet actuellement une forte évolution (voire mutation) avec d'une part les grilles de calcul et d'autre part, les multiples appareils mobiles intégrant des systèmes informatiques de plus en plus performants et complexes. Ces systèmes informatiques intégrant un parallélisme massif ou/et une mobilité des composants sont un défi pour le génie logiciel qui doit apporter de nouvelles méthodes et des outils de production de logiciel pour la description de l'architecture de ces systèmes complexes et pour leur validation et/ou certification. En plus, la montée en puissance des techniques en géomatique va permettre de prendre une bonne décision à partir des données spatiales et temporelles dans différents domaines.

Chapitre 2 : Présentation du Madagascar National Parks Parc National Isalo

Le Parc National Isalo présente une beauté ancestrale unique due à ses vestiges archéologiques, ses sanctuaires culturels « Bara »(en particulier ses sites funéraires) et ses massifs ruiniformes sis dans une atmosphère magique faite de violents contrastes entre :

- L'aridité et le silence des falaises ocre où pousse une rare végétation rupicole (divers espèces de pachypodes, d'aloès, et d'euphorbes) écrasée par le soleil,
- Les appels de lémuriens (*Prophithecus verreauxi*, *Lémur fulvus*, *Lémur Catta*), des chants d'oiseaux (une cinquantaine d'espèces) et le doux murmure de l'eau qui résonnent dans les oasis de verdure le long des rivières et au fond des canyons.

2.1 Localisation

La figure 2.1 montre la carte de localisation du parc National de l'Isalo



Figure 2.1 : Carte de localisation du Parc National de l'Isalo

Le parc Nationale Isalo est à 683 Km au Sud-Ouest de Tananarive, environ 269 Km au Sud de Fianarantsoa et 243 Km au Nord-Est de Tuléar.

Il se trouve à la province de Fianarantsoa, préfecture d'Ihosa et au Commune rurale de Ranohira.

Et pour la géolocalisation : 22°22 à 22°40 de latitude, 5°23 de longitude et 514 à 1268 Km d'altitude. Il est limité au Sud par la RN7, son massif ruiniforme à l'Ouest, la rivière de Menamaty à l'Est et le massif lui-même au Nord.

2.2 Historique

Le PNI a vu naissance le 19 Juillet 1962 régit par le décret n°62.731, et c'était le WWF qui occupait sa gestion jusqu'à ce que l'Entreprise ANGAP a pris le relais.

Depuis 1995, il a été sous contrôle de l'ANGAP qui a un dessein de protéger Isalo parce qu'il est une richesse exclusif à Madagascar et de conserver ses biodiversités, mais l'Entreprise a été nommé sous le titre de MNP à partir de l'année 2008 et jusqu'à aujourd'hui.

2.3 Rôle et mission

Le MNP concourt avec un effort accentué pour établir, conserver et gérer de façon plus chronique d'un réseau national du Parc et réserve représentatif de la biodiversité biologique et du patrimoine naturel propre à Madagascar.

- Il assure la planification du développement rural ciblé dans les zones circulaires.
- Il est chargé d'assister l'encouragement de la recherche sur la biodiversité et le progrès de l'écotourisme ainsi que la mobilisation des fonds nécessaires à la concrétisation de toutes ces organisations de projets.
- Il coordonne les formations et les informations à tous les niveaux, et organise les activités de conservation au sein des Aires Protégées.

2.4 Profil des Postes

2.4.1 Le Directeur du Parc ou CUG

Le Directeur du Parc est à la fois le premier responsable de son unité (l'Entreprise) et le représentant du siège dans toutes les circonstances. En tant qu'à la tête de l'Unité, il est donc évidant de prendre en charge de :

- Veiller à la protection du Parc par le biais de la réalisation intégrale des activités techniques, suivant la planification, les normes et les référentiels en vigueur, en impliquant la population riveraine, les autorités locales, régionales et tous les partenaires.

-
- Veiller à la promotion du Parc et à l'optimisation des ressources (humaines, matérielles et financières) en vue de la pérennisation financière.
 - Contrôler la gestion administrative et fiscale, financière et aussi l'audit à l'intérieur de son Unité

2.4.2 Le Chef de Volet Administratif et Financière

En respect des procédures et règlements en vigueur, le chef de volet administratif et financière est à l'incommodité de :

- Veiller à la disponibilité et à la sécurisation des actifs et du patrimoine de MNP au niveau du Parcs.
- Assurer la production des informations financières fiables et exhaustives dans les délais impartis.
- Assurer la gestion administrative du personnel
- Assurer l'éligibilité des dépenses en collaboration avec le CUG.

2.4.3 Le Chef de Volet d'Appui au Développement et Education Environnementale(CVADEE)

Le CVADEE occupe tous les responsabilités concernant l'éducation environnementale et aussi à l'assurance de :

- Veiller à l'application du cadre de cogestion de MNP au niveau de l'UG,
- Veiller à ce que le Parc réalise ou facilite entièrement dans le but d'un développement économique intégré, les différentes actions alternatives prévues dans le PAG et dans les documents cadres de développement local et régional,
- Veiller à ce que la communication sur l'éducation environnementale et la confiance mutuelle règnent entre les communautés locales (COSAP, CLP etc...) et les gestionnaires d'AP en vue de promouvoir l'attitude favorable à l'environnement,

2.4.4 Le Chef de Volet de l'Ecotourisme (CV Ecot)

Il occupe une très grande responsabilité envers les clients, les opérateurs partenaires et surtout à l'évolution éco touristique. Il est à la charge de:

- Installer les aménagements, installations et matériels adéquats pour le nombre de touristes accueillis,

-
- Etablir une coopération excellente entre les gestionnaires et les opérateurs touristiques pour améliorer le séjour des visiteurs et pour protéger les valeurs,
 - Assurer la qualité de l'accueil au niveau du parc,
 - Collecter et mettre à jour les bases de données relatives au développement du volet tourisme selon instructions Marketing et services produits,
 - Elaborer avec l'appui du SAT et du Directeur de Parc le plan d'écotourisme du Parc et le met à jour,
 - Elaborer le plan d'interprétation et le met à jour,
 - Elaborer le plan de masse de chaque zone exploitée et / ou à aménager,
 - Elaborer le plan d'évacuation des visiteurs en cas de force majeure,
 - Etablir les fiches d'identification de chaque ouvrage et le plan d'entretien à suivre par an,
 - Etablir les plans d'exécution de chaque ouvrage situés le long des circuits touristiques,
 - Mettre en place les sites de camping, latrines, panneaux de signalisation et autres...
 - Collecte les informations (données, images, films..) sur l'activité touristique du Parc, et leurs impacts sur l'économie locale et la satisfaction des visiteurs
 - Etudier, proposer et collecter les données/informations nécessaires pour la production des supports de promotion et les informations utiles aux visiteurs du Parc
 - Appuie les groupements locaux voulant entreprendre en écotourisme (petits réceptifs, producteurs de produits locaux et artisanaux...).

2.4.5 Le Chef de Volet de Conservation et Recherche

Il veille à ce que les systèmes de protection soient totalement efficaces :

- par la mise en œuvre des activités relatives à la protection du parc,
- par le suivi de la santé écologique du Parc,
- par la coordination des activités de recherche,

Il constitue une documentation locale sur toutes les cibles de conservation du Parc.

2.4.6 Le Chef Secteur

- Assurer la protection de son secteur et des investissements y réalisés pour la pérennisation,
- Assurer que l'ensemble des différentes catégories de la population et des autorités locales soutienne le Parc au niveau des villages de son secteur,
- Identifier les aptitudes techniques particulières et les besoins en formation de ses AGP, et les présente pour le plan de formation,

2.4.7 Le Secrétaire

- Assurer le classement des pièces comptables par opération et les travaux de comptabilité élémentaires,
- Assister le chef de Volet Administratif et Financier sur les arrêtés périodiques de compte,
- Assister les chefs de volet et chefs de secteur dans l'établissement des DED,
- Assurer le secrétariat administratif de l'Unité de gestion.

2.4.8 Responsable de Centre d'Interprétation

- Appuyer la lutte contre les pressions en cherchant des palliatifs,
- Collecter des informations pour une meilleure connaissance scientifique du parc,
- Suivre l'application des conventions de recherche sur place,
- Veiller au confort logistique des chercheurs utilisant le centre de recherche : tentes, matériels divers,
- Aider le chef de volet dans le traitement des données de suivi écologique,
- Tenir une documentation de base sur la flore et la faune et les recherches effectuées au site,
- Participer à la collecte de données nécessaires à l'interprétation des richesses du parc.

2.4.9 Agents du Parc

- Assurer l'exécution des travaux sur terrain du parc selon les nécessités identifiées par les responsables et les spécialités des agents et rapporte en conséquence,
- Définir par discussion avec le chef de secteur ses objectifs individuels annuels

- Réaliser les travaux de surveillance, de patrouille et de contrôle de l'AP avec rapport des délits perpétrés au niveau du secteur de rattachement.

2.4.10 Hôtesse d'Accueil

- Assurer l'accueil de la clientèle : proposition de services de visite et guidage correspondant,
- Assurer la vente des tickets,
- Assurer la vente des articles et marchandises de l'éco shop,
- Assurer la location de matériels de campement,
- Tenir les livres de compte,
- Effectuer le versement en fin de journée des recettes DEAP auprès du CVAF,
- Assurer la base de données relative à la rubrique du volet,
- Collecte les informations sur la clientèle et mise à jour de la base de données.

2.4.11 Logisticien

- Tenir des fiches de stocks et magasinages,
- Enregistrer des mouvements des stocks, sortie des articles,
- Inventaire périodique,
- Gestion des mouvements des matériels roulants et/ou navigants de l'UG,
- Suivi des diverses échéances relatives aux véhicules roulants et /ou navigants,
- Assurer la gestion des stocks, des mouvements des biens.

2.5 Organigramme du MNP PNI

L'organigramme du Madagascar National Parks Parc National Isalo se représente dans la figure 2.2.

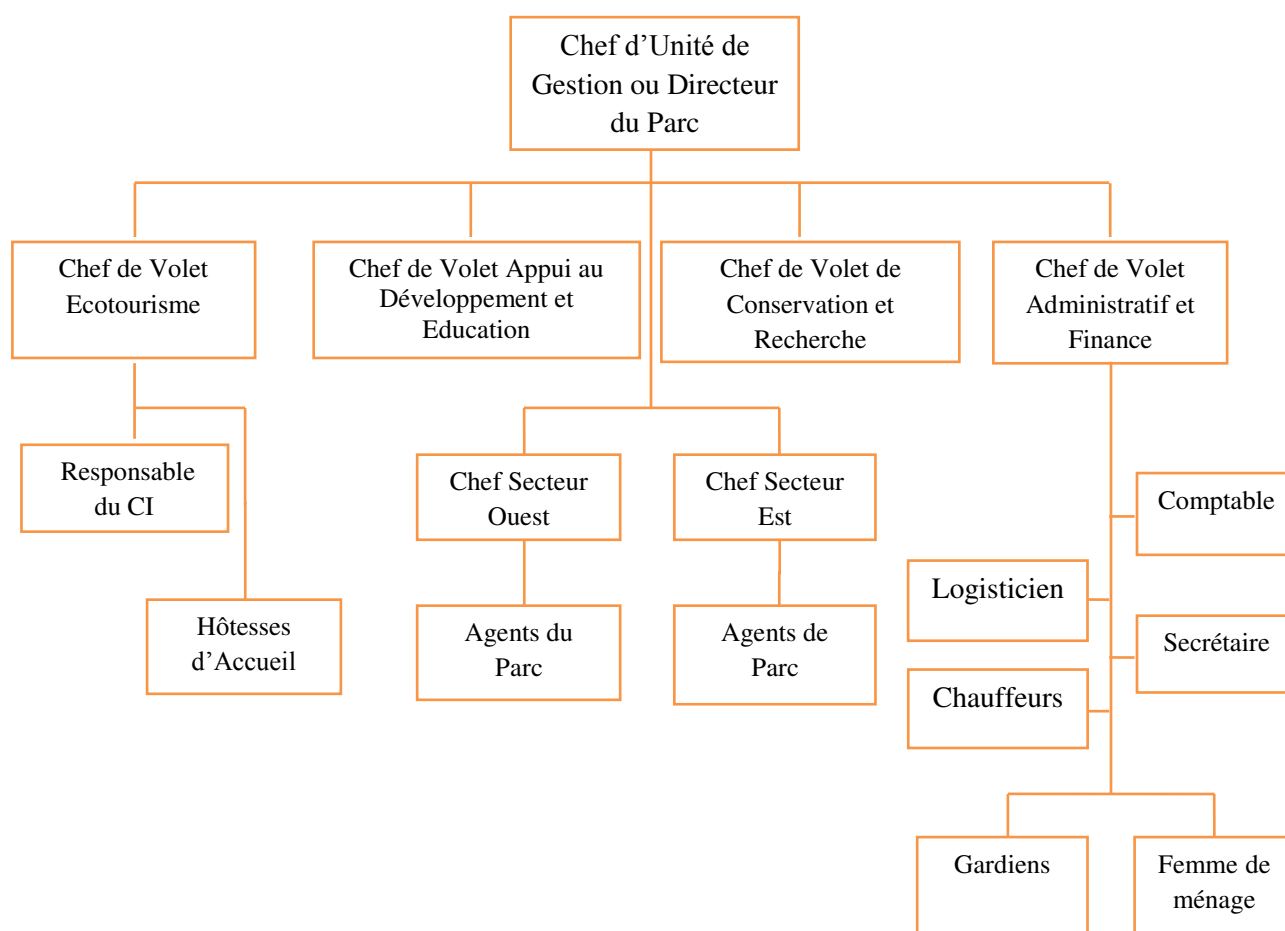


Figure 2.2 : Organigramme du MNP PNI

Chapitre 3 : Description du projet

3.1. Cahier des charges

3.1.1. Définitions

Le « Cahier des charges » [Pierre G., 2005] [Bakari M., 2016] est un document qui est attendu du maître d'œuvre par le maître d'ouvrage. Il s'agit d'un document de façon plus précise avec un vocabulaire simple. Dans la mesure où le maître d'œuvre est compétent pour proposer une solution technique, le cahier de charge fait apparaître les besoins d'une manière fonctionnelle, indépendamment de toute solution technique. Il s'agit aussi d'une part un document permettant de garantir au maître d'ouvrage que les livrables seront conformes à ce qui est écrit et d'autre part, le maître d'ouvrage modifie son souhait au fur et à mesure du projet et demande au maître d'œuvre des nouvelles fonctionnalités non prévues initialement.

3.1.2 Contexte et objectifs

L'assidu apprentissage rétablis dans le cadre de la présentation de mémoire de fin de cycle en vue de l'obtention d'un diplôme de « Licence professionnelle en Développement d'Application et Système d'Information » au sein de l'EMIT qui nous agréé d'organiser la vie professionnelle.

L'hôtesse qui est responsable d'accueil fait face au problème de répétition de travail manuellement et machinalement, et à l'incertitude de calcul du recette du jour en fin de justifier le stock de ticket que l'accueil a pour le reste du vente de la journée et la recette qu'elle a virée, ce qui veut dire de faire l'état de stock et du vendu pour le PCA en une version physique. Pour les aider, quelle est la possibilité pour résoudre leur problème? Comment faire un calcul automatique pour avoir la pièce comptable d'accueil ou la recette D.E.A.P? Comment enregistrer les données dans une base de données sécurisée?

D'après une discussion que nous avons faite avec les responsables d'accueil, elles ont besoins une application capable d'envoyer un message d'information pour informer le guide ou leur chef ou aux agents du parc et notamment de faire un calcul automatiquement pour le PCA et la note de débit. Et pour leur Chef, comment faire, sans passer au triage des données comme dans l'Excel qu'il utilise aujourd'hui, pour avoir la statistique mensuelle et annuelle des données? De faire le triage automatiquement ?

Alors, le but de ce présent projet est que l'application se déroule convenablement, de conserver ces fonctionnalités attendus dans l'application, d'améliorer la gestion périodique afin d'éviter l'incertitude et la perte de temps.

3.1.3 Les acteurs du projet

Les acteurs de ce projet sont le Maitre d'œuvre et le Maitre d'ouvrage.

3.1.3.1. Maitre d'ouvrage

Le maître d'ouvrage (parfois maîtrise d'ouvrage, notée MOA) est une entité porteuse du besoin, définissant l'objectif du projet, son calendrier et le budget consacré à ce projet. Le résultat attendu du projet est la réalisation d'un produit, appelé ouvrage.

3.1.3.2. Maitre d'œuvre

Le Maitre d'œuvre est une personne morale garante de la bonne réalisation technique des solutions. Le maître d'œuvre (ou maîtrise d'œuvre, notée MOE) est aussi une entité retenue par le maître d'ouvrage pour accomplir l'ouvrage, dans les conditions de délais, de qualité et de coût fixés par ce dernier conformément à un contrat.

3.1.4 Résultats attendus

C'est pour avoir une application qui peut :

- gérer instinctivement le calcul de la vente du jour ;
- gérer automatiquement l'état de stock de la marchandise ;
- être capable d'afficher les statistiques de la vente soit hebdomadaire, soit mensuelle ou annuelle ;
- envoyer un ou plusieurs messages mobiles pour informer les guide en cas d'urgence ou s'il y a de client qui n'a pas encore un guide ;
- chercher un guide et son profil ;
- afficher le profil d'un site au parc Isalo...

3.1.5 Activités et tâches

L'étude sollicite la réalisation des diverses enthousiasmes à savoir :

- Une recherche et analyse documentaire afin d'unir les informations vacant sur la conception et réalisation de cet application web;
- Des enquêtes auprès du garant des services d'accueil ou bien l'hôtesse et au responsable des volets écotourisme;
- Exploitation des résultats ;
- Elaboration des tentatives de transformation.

3.1.6 Déroulements du stage

Comme tout début d'un stage, on a commencé le nôtre le 10 Décembre 2016 avec une durée conventionnée de trois (03) mois dès notre arrivée au sein de la société. Ce formation qu'on a fait au service d'accueil du parc national Isalo nous a permis de savoir sur quoi elles ont

travaillé et la difficulté qu'elles subissent tout au long de leur séjour du travail qui nous conduit à proposer des solutions possibles. Pendant notre stage nous avons effectué des entretiens au niveau des responsables pour obtenir des informations. Les données nécessaires pour la rédaction de notre mémoire sont complètes, grâce au bon accueil des différents responsables.

Le tableau 3.1 suivant nous montre les déroulements du stage.

	Décembre		Janvier				Février				Mars				Avril	
Activités	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2
Délimitation de thème	X															
Discussion et choix du projet		X														
Mise au point du projet			X													
Recherche documentaire			X	X	X											
Rencontre des personnes				X	X											
Enquêtes et entretiens				X	X											
Rédaction du mémoire					X	X	X		X	X	X	X				
Conception et réalisation							X	X	X	X	X	X				
Elaboration des essais												X	X			
Préparation de soutenance														X	X	
Soutenance de mémoire																X

S1 : Semaine 1, S2 : Semaine 2, S3 : Semaine 3, S4 : Semaine 4

3.2. Formulations

Le MNP Parc National Isalo est un établissement semi-privé à caractère administratif ainsi, ce projet de mémoire parle notamment d'une partie de son service.

Pour parvenir aux besoins de l'utilisateur de cette application, nous avons fait au moment convenable des interviews auprès des responsables d'accueil et de leur supérieur hiérarchique le chef de volet écotourisme.

Le présent projet a pour objectif principaux de :

- mettre en place un système sécurisé ;
- mettre en place une interface graphique simple et intuitive ;
- gérer automatiquement le PCA ou la recette DEAP ;
- répondre aux besoins de son utilisateur (hôtesses d'accueil) ;
- élaborer une application web permettant de gérer automatiquement leur vente.

3.3 Besoins des utilisateurs

Principalement les nécessités de l'utilisateur paraient avoir :

- Une application bien sécurisée ;
- Seul le responsable qui peut avoir l'autorisation d'accès à la base de données ;
- Une application qui peut communiquer avec le téléphone mobile ;
- Une application qui suit la technologie moderne de nos jours.

Ces besoins principaux peuvent être classés en besoins fonctionnels et non fonctionnels.

3.3.1. Besoins fonctionnels

Les besoins fonctionnels expriment une **action** que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées). Ils ne deviennent opérationnels que s'ils les désaltèrent. L'application doit couvrir principalement les besoins fonctionnels suivants :

3.3.1.1 Ajout, Modification et Suppression des données

Une base de données permet de mettre des informations à disposition d'utilisateurs pour qu'ils puissent les consulter, les modifier, les mettre à jour... Dans la très grande majorité des cas, ces informations sont très structurées, et la base est localisée dans un même lieu et sur un même support. Donc, pour pouvoir insérer ou modifier ou supprimer un certain nombre de données, il nous faut un formulaire d'ajout, de modification et de suppression pour ça. En plus, c'est une action qui va initialiser les contenue de la base.

3.3.1.2 Automatisation de calcul

Le MNP est classifié comme une grande entreprise, le travail est physiquement indispensable et a besoins de rapidité sans faire des erreurs, sinon on remboursera le manque. Donc, pour éviter à des erreurs comme ça, il nous faut une application qui peut agir selon nos besoins de calculer la recette ou l'addition automatiquement.

3.3.1.3 Stockage des données dans une base de données

Toutes les informations qui sont apparus sur le formulaire de l'application sont stockées dans une base de données, c'est pour servir la mise à jour et la recherche des données.

3.3.1.4 Envoie de message

La technologie ne cesse vraiment jamais d'évoluer, alors la future application permettra son utilisateur d'envoyer des informations à partir de l'application sans utiliser un téléphone pour les envoyer.

3.3.2. Besoins non fonctionnels

Tous les systèmes d'information à un certain point dans leur cycle de vie qui doivent considérer des besoins non-fonctionnels et leurs tests. Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation [MVOU-OSSIALAS Wilfried-Erisco., 2003].

3.3.2.1. Exigence de qualité

Pour une facilité d'apprentissage, l'interface graphique de l'application doit être simple et intuitive. Les formulaires à remplir doivent utiliser des termes techniques métier et présentent certains choix pour les valeurs fréquentes.

3.3.2.2. Exigence de performance et sécurité

L'application devra être hautement sécurisée car les informations ne devront pas être accessibles à tous les personnels sauf à son utilisateur spécifié qui sont l'hôtesse d'accueil et leur chef de volet. Les recherches effectuées sur l'application ne doivent pas prendre beaucoup de temps, ainsi que le calcul. Toutes activités sur l'application doivent être facilement atteintes.

3.3.2.3. Contrainte de conception

Les services de l'hôtesse d'accueil sont tous pratiquement faites avec le tableur Excel qui fonctionnent correctement et avec le papier aussi pour les stocker physiquement mais l'entreprise a besoins une nouvelle technologie qui pourrai faciliter et capable d'imprimer leur données sans passer par le travail manuellement pratique avec le stylo de leur travaux.

Les pièces importées a l'environnement de mise en examen, utilisées pour l'aide à la détection d'anomalies, ne seront pas enregistrées dans la base de donnée car leurs existences ne seront que temporaire.

Donc, le système doit être développé avec la technologie web comme système de gestion de base de données relationnelle et avec une langage et framework pour soutenir leur travail.

3.4 Moyens pour la mise en œuvre

Pour exécuter ce projet, les ressources humaines et matérielles, et notamment logiciel sont importants.

3.4.1. Ressources humaines

Dans notre projet, les acteurs sont regroupés en trois (03) : la maitrise d'ouvrage, la maitrise d'œuvre et les utilisateurs.

- La maitre d'ouvrage est représentée par le Chef de Volet Ecotourisme qui a pour rôle de donner les besoins et leurs spécifications détaillée ;
- Les utilisateurs finaux, qui sont les deux hôteses qui utilisera l'application et ne verront le système que lors de sa mise en exploitation ;
- Les utilisateurs référents, qui sont impliqués dès l'initialisation du projet, sont représentés par le Chef des volets de l'entreprise et qui sont le maitre d'œuvre.

3.4.2. Ressources matérielles et logicielles

Pour la réalisation du projet, les moyens matériels et logiciels nécessaires sont :

Un ordinateur utilisé comme poste de travail installé les applications suivantes :

- Un logiciel de gestion de projet ;
- Un logiciel de modélisation ;
- Un logiciel de suite bureautique pour créer les documents et rapports ;
- Un environnement de développement intégré ;
- Une connexion internet pour se documenter.
- Un ou plusieurs serveurs hébergeant les services suivants :
 - ✓ Serveur d'application ;
 - ✓ Serveur de base de données.

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4 : Présentation du modèle de cycle de vie et du langage de modélisation utilisés

4.1. Cycle de vie d'un logiciel

4.1.1. Définition d'un modèle

Un modèle est une abstraction de la réalité [Anne-Marie H., © 19/12/02 2-13,] [Hajalalaina A. R., 2016]. Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité, en vue d'une utilisation précise. L'abstraction désigne aussi le résultat de ce processus, c'est-à-dire l'ensemble de la caractéristique essentielle d'une entité, retenue par un observateur. Un modèle est une vue subjective mais pertinente de la réalité. Il définit une frontière entre la réalité et la perspective de l'observateur. Ce n'est pas « la réalité », mais une vue très subjective de la réalité. Bien qu'un modèle ne représente pas une réalité absolue, il reflète des aspects importants de la réalité, il en donne donc une vue juste et pertinente. Le caractère abstrait d'un modèle doit notamment permettre :

- de soutenir la compréhension du système étudié : il réduit la complexité du système étudié ;
- de simuler le système étudié : il représente le système étudié et reproduit ses comportements.

4.1.2. Modèle de cycle de vie d'un projet web

Comme toute application informatique, une application web nécessite d'être défini, puis conçu, avant d'être réalisé, sans oublier qu'il existe aussi un « après projet », qui devra être préparé : exploitation, promotion et animation.

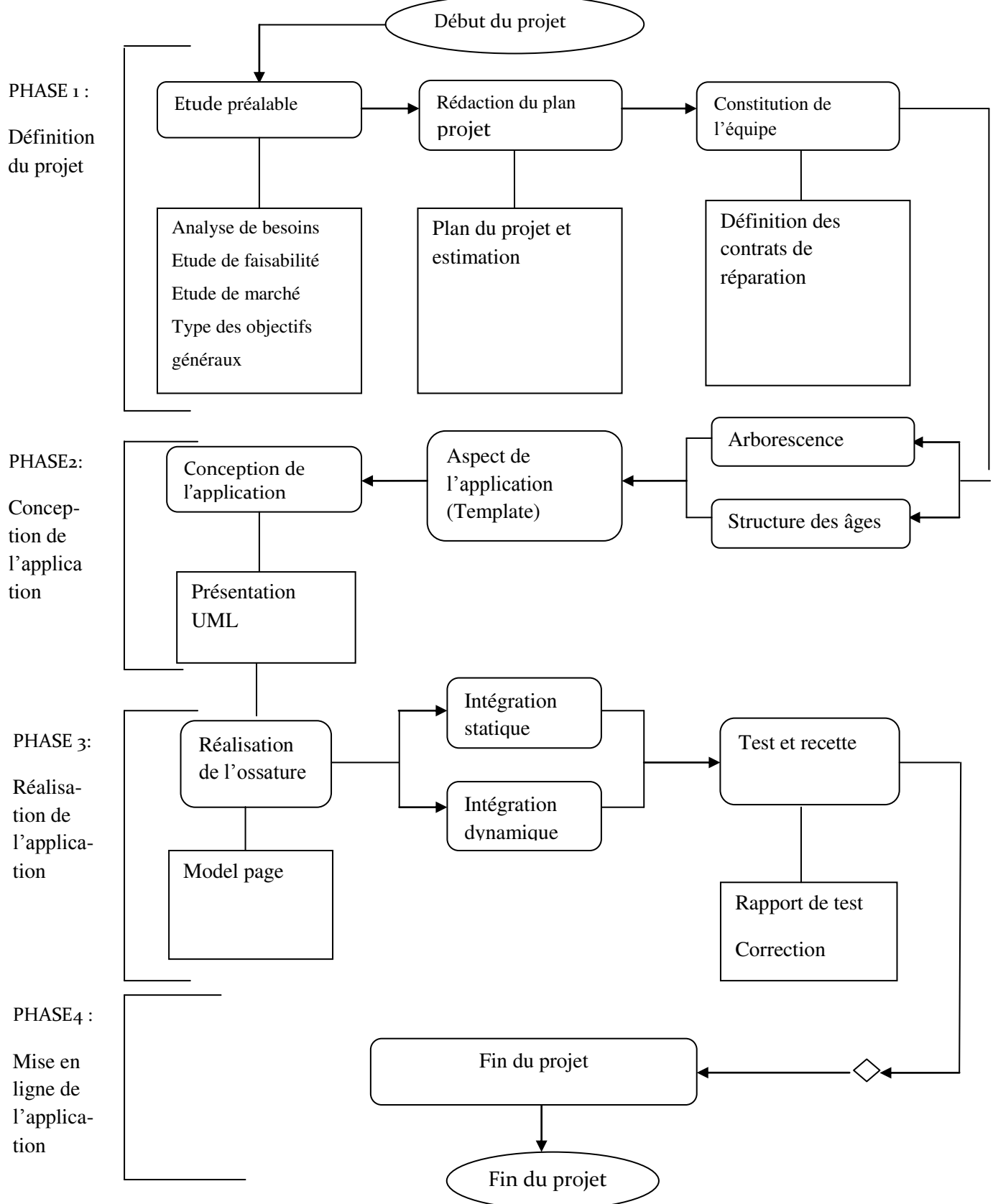


Figure 4.1 : Modèle de cycle de vie d'une application web [adapté à partir du modèle de cycle de vie d'une application web par Dina N., 2014]

4.1.3. Modèle de cycle de vie d'un logiciel

Dans le domaine de l'informatique et notamment dans le domaine du développement des logiciels, on entend par *cycle de vie d'un logiciel* une procédure en vue du développement dudit logiciel de sa conception à sa disparition.

En effet, le cycle de vie d'un logiciel est la description d'un processus couvrant les phases de:

- Création du logiciel ?
- Distribution du logiciel sur un marché ?
- Disparition du logiciel.

Ainsi, l'objectif du cycle de vie d'un logiciel est de définir des repères intermédiaires permettant la validation du développement logiciel, c'est-à-dire :

- la conformité du logiciel avec les besoins exprimés,
- la vérification du processus de développement.

Le cycle de vie d'un logiciel permet donc de détecter les erreurs au plus tôt et ainsi de maîtriser :

- la qualité du logiciel,
- les délais de sa réalisation,
- les coûts associés,
- les risques.

En tant que procédure, le cycle de vie d'un logiciel est composé de différentes étapes. Ainsi, en général, le cycle de vie d'un logiciel comprend au minimum les étapes suivantes :

- *Analyse des besoins*, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
- *Conception globale*, il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
- *Conception détaillée*, consistant à définir précisément chaque sous-ensemble du logiciel.
- *Codification* (Implémenter ou programmer), soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- *Tests unitaire*, permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.
- *Recette*, c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.

- *Intégration*, dont l'objectif est de s'assurer de la liaison des différents éléments (modules) du logiciel. Elle fait l'objet de *tests d'intégration* consignés dans un document.
- *Documentation*, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- *Déploiement ou mise en production*,
- *Maintenance et évolution*, comprenant toutes les actions correctives et évolutives sur le logiciel.

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement. En effet, il existe plusieurs MCV. Il est donc important pour une équipe de développement de logiciels de bien choisir son MCV. Car, ce dernier prend en compte, en plus des aspects techniques, l'organisation et les aspects humains. Ainsi, chaque équipe de développement de logiciels se doit de comparer les différents MCV en vue de choisir celui qui est adapté à leur contexte.

4.1.4. Les différents modèles de cycle de vie

Il existe une multitude de MCV (en cascade, en V, par incrément, en spirale, etc.). Face à cette multitude de modèles, nous avons fait le choix de faire la comparaison sur quatre (04) cycles de vie à savoir :

- Le modèle en Cascade ;
- Le modèle en V ;
- Le modèle en Spirale ;
- Le modèle par Incrément ou itératif.

4.1.4.1. Cycle de vie en Cascade

Le *modèle en Cascade* est hérité de l'industrie du BTP (c'est-à-dire de l'industrie lourde). Ce modèle repose sur les hypothèses suivantes :

- on ne peut pas construire la toiture avant les fondations ;
- les conséquences d'une modification en amont du cycle ont un impact majeur sur les coûts en aval.

Les phases traditionnelles de développement sont effectuées simplement les unes après les autres, avec un retour sur les précédentes, voire au tout début du cycle. Le processus de développement utilisant un cycle en cascade exécute des phases qui ont pour caractéristiques :

- de produire des livrables définis au préalable ;

- de se terminer à une date précise ;
- de ne se terminer que lorsque les livrables sont jugés satisfaisants lors d'une étape de validation-vérification.

Pour mieux comprendre ce modèle, on peut dire qu'il a une facilité de comprendre, d'utiliser et adapter à une équipe inexpérimentée. Les limites de chaque étape sont visibles. En plus, ce modèle facilite un management du projet et la définition des besoins est non-évolutive. Mais, tous les besoins doivent être bien spécifiés au départ. Il donne une fausse impression de l'avancée des travaux, pas d'interaction entre les phases de développement, pas de retour en arrière d'une phase à l'autre et l'intégration n'a lieu qu'à la fin du cycle, donc le client peut se retrouver non satisfait.

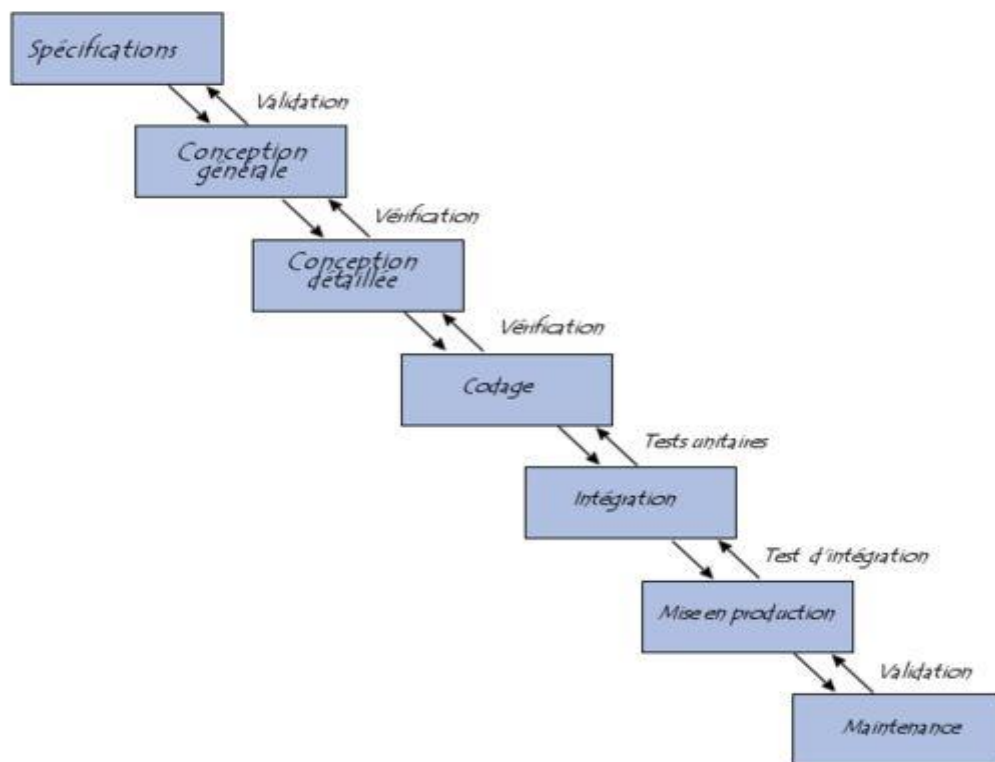


Figure 4.2 : Modèle en cascade [Wikipédia/cycle en cascade]

4.1.4.2. Cycle de vie en V

Le *modèle du Cycle en V* a été imaginé pour pallier le problème de réactivité du modèle en cascade. Ce modèle est une amélioration du modèle en cascade qui permet en cas d'anomalie, de limiter un retour aux étapes précédentes. Ses phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer le logiciel. De plus, le cycle en V met en évidence la nécessité d'avancer et de préparer dans les étapes descendantes les « attendus » des futures étapes montantes : ainsi les attendus des tests de validation sont définis lors des spécifications, les attendus des tests unitaires sont définis lors de la conception. Mais tous les tests sont effectués et celui du contrôle se fait progressivement à chaque étape. Et son phase de validation sont prises en main très tôt dans le processus de développement. Or, le processus n'est pas itératif et une mauvaise prise en compte des événements concurrents et des changements de la spécification des besoins. En plus, il ne contient pas les activités d'analyses de risques. Malgré cela, il est excellent pour les systèmes requérant une grande sûreté.

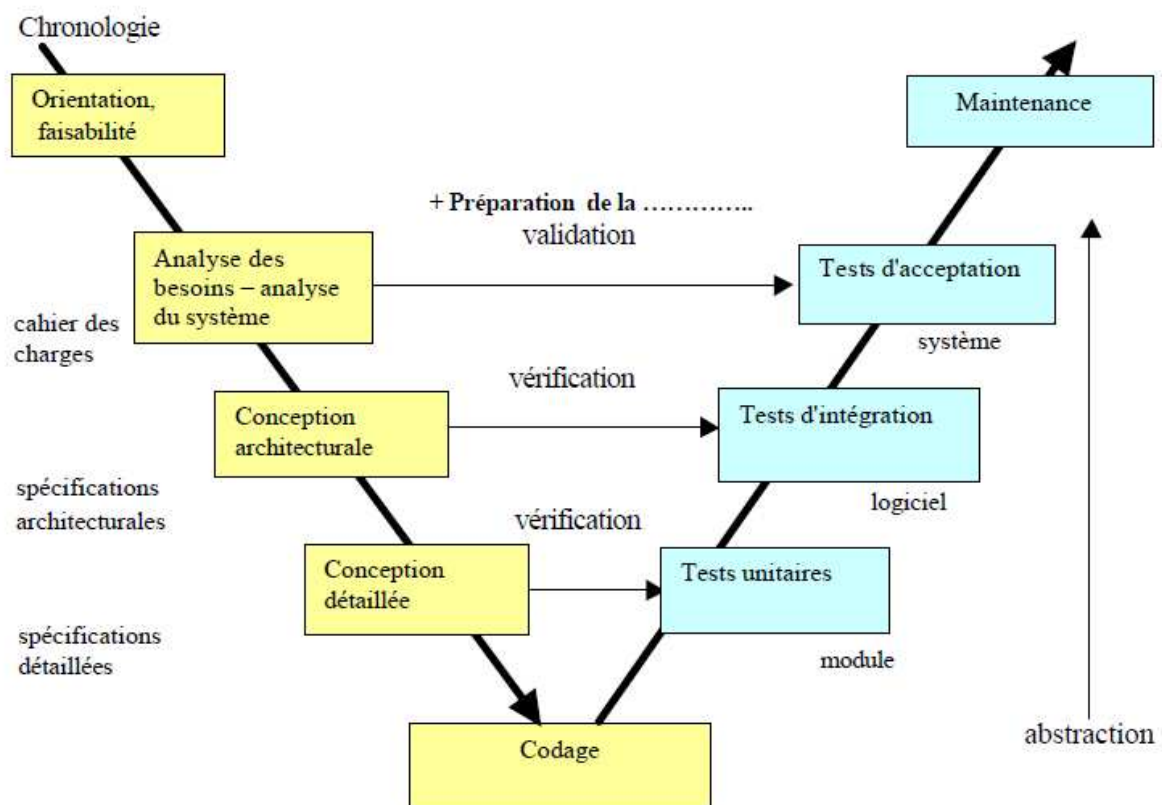


Figure 4.3 : Schéma de modèle en V [Conception informatique/cycle en V]

4.1.4.3. Modèle de cycle de vie en spirale

Le développement reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet et robuste. Le cycle en spirale met cependant plus l'accent sur la gestion des risques que le cycle en V. En effet, le début de chaque itération comprend une phase d'analyse des risques. Celle-ci est rendue nécessaire par le fait que, lors d'un développement cyclique, il y a plus de risques de défaire, au cours de l'itération, ce qui a été fait au cours de l'itération précédente. Alors, les fonctions critiques à haut risque sont développées en premier lieu, la conception ne doit pas forcément être terminée. Son développement se fait en interaction avec les clients et l'évolution du coût est sous contrôle. Mais le temps consacré à l'évaluation des risques est trop élevé pour des petits projets. De plus, le temps mis à planifier, fixer les objectifs, les prototypes peut être excessif et la spirale peut être infinie. Donc, il est difficile de définir les objectifs et les points de validation intermédiaires entre les différentes étapes.

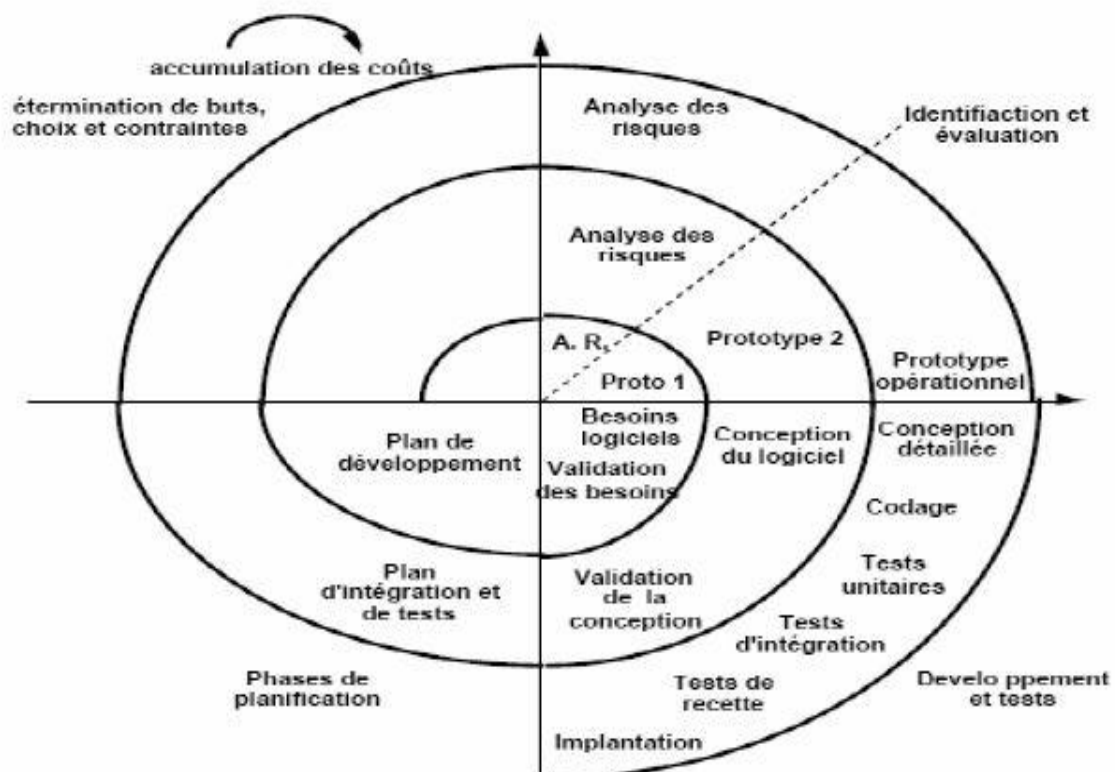


Figure 4.4 : Modèle en spirale [Wikipedia.org/wiki/spiral_model]

4.1.4.4 Modèle de cycle de vie par incrément ou itératif

Le cycle de vie par incrément, on sépare les activités des artefacts. Un artefact étant le produit issu d'une activité. Ainsi, on applique un cycle de type roue de Deming sur la production d'une documentation, d'un composant, d'un test, etc.

Rapportée à une activité de type gestion de projet, la première phase sera celle de

- la faisabilité : l'acceptation d'un nouveau besoin ;
- l'élaboration : on imagine comment on va le réaliser ;
- la fabrication : construction ;
- la transition : tout est mis en œuvre pour livrer au client.

L'avantage d'utiliser ce modèle est que le client peut valider chaque étape de processus. La délivrance du produit est rapide c'est-à-dire qu'un produit exploitable peut être délivré à tout moment et le risque du changement des besoins est minimal. Mais l'inconvénient, ce modèle requière une bonne planification, une bonne conception et la définition complète des fonctionnalités du système pour une définition des différents incréments. Et les différentes interfaces doivent être bien définies.

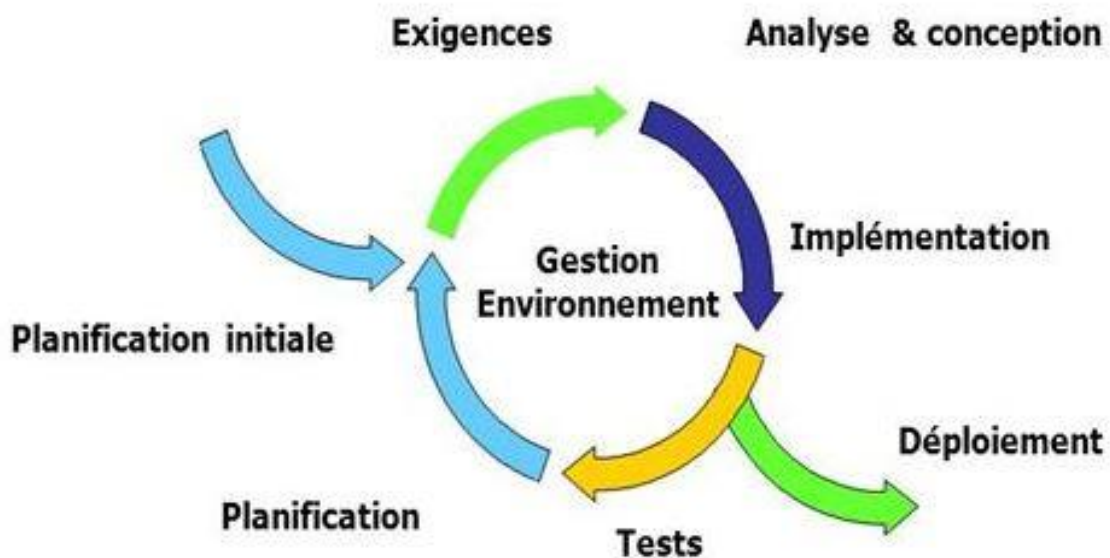


Figure 4.5 : Modèle par incrément ou itératif

Chaque itération a pour finalité d'une version exécutable.

4.1.4.5 Cycle de vie incrémentale

Dans les modèles précédents, un logiciel est décomposé en composants développés séparément et intégrés à la fin du processus. Dans les modèles par incrément un seul ensemble de composants est développé à la fois : des incréments viennent s'intégrer à un noyau de logiciel développé au préalable. Chaque incrément est développé selon l'un des modèles précédents. En bref, le modèle par incrément découpe le système en domaines qui sont traités individuellement sur le modèle en cascade.

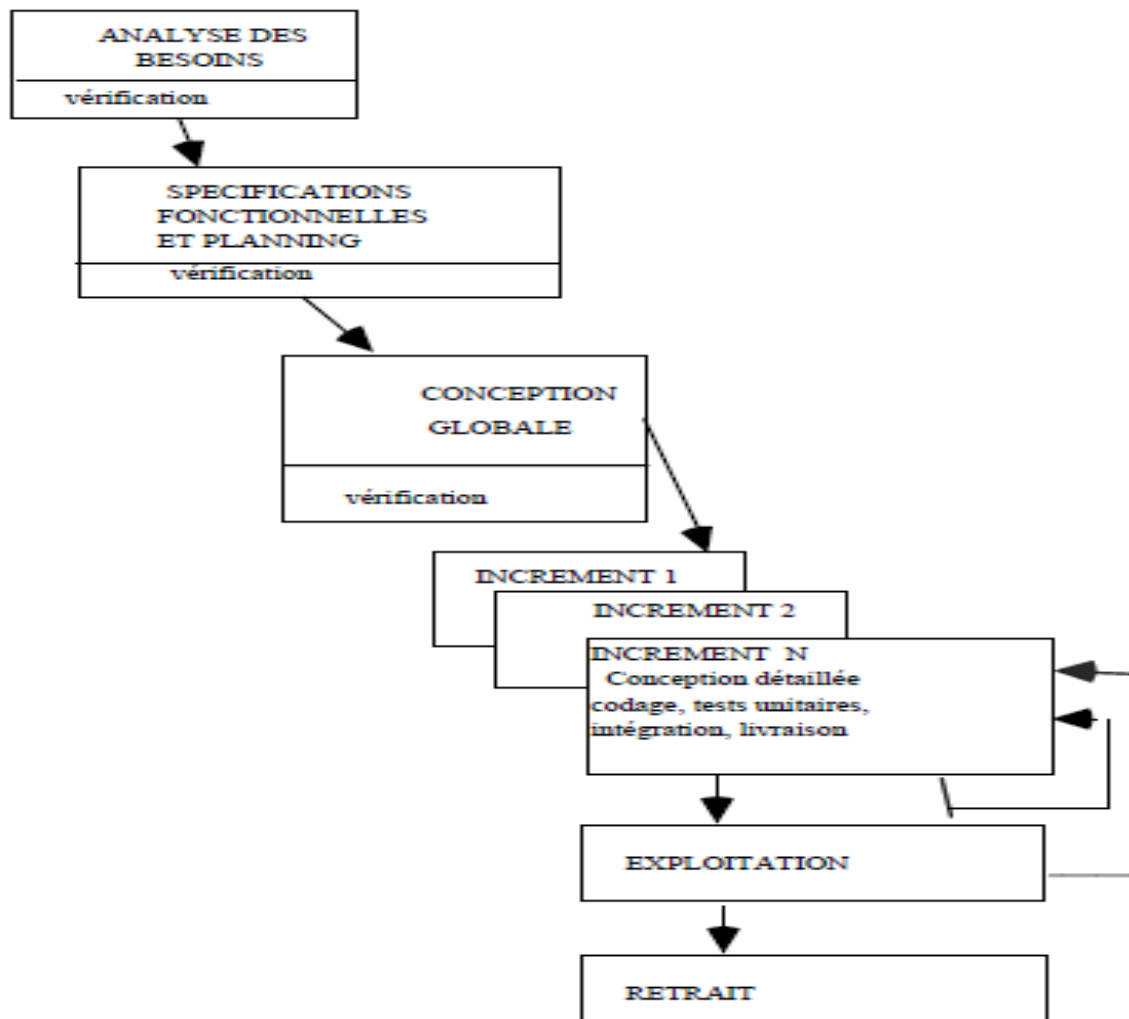


Figure 4.6: Modèle incrémentale

4.1.5. Comparaison de modèle de cycle de vie

Nous venons de faire la description de ces cinq (05) MCV. Nous allons maintenant faire la comparaison entre les quatre (04) premiers selon le critère du moment d'utilisation.

En parlant du modèle en cascade, on l'utilise quand la phase de spécification a été très bien faite, quand la définition du produit est stable. Il ne s'agit d'une nouvelle version d'un produit existant. En plus, quand l'implantation d'un produit existant sur une nouvelle plateforme et une bonne maîtrise de la technologie. Mais pour le modèle en V, les spécifications de besoins doivent être bien faites, la solution à développer et la technologie à utiliser doivent être connues. Et les changements doivent être faits avant l'analyse.

Si cela qu'on a besoin pour les deux modèles, quels sont donc les deux suivants ? On utilise le modèle en spirale quand les coûts et l'évaluation des risques sont importants. Et aussi pour des projets à risque au moins moyennement élevé et à long terme dont les financements peuvent varier. En outre, quand les utilisateurs ne définissent pas clairement leurs besoins et que la spécification des besoins soit complexe. Et en fin, pour le modèle par incrément, on l'utilise sur un projet utilisant de nouvelles technologies, sur des projets ayant une durée de développement assez longue. Quand la plupart des fonctionnalités doivent être connues au départ mais certaines ne seront utilisées que plus tard et que les risques, le financement, la complexité du logiciel, l'élaboration ou les besoins pour les bénéfices sont maximum dès le départ.

4.1.6. Choix et justification

En disant comme résultat après avoir les comparer, il n'est pas facile de prendre une décision concernant ces différents cycles de vie. Chacun a ses forces, ses faiblesses et un cadre d'utilisation bien déterminé. Malgré tout, nous constatons une plus grande utilisation du Cycle en V par la plupart des équipes de développement.

Les principaux avantages de l'utilisation du modèle en V sont les suivants :

- Il permet l'organisation du travail et des équipes ;
- Il favorise la décomposition hiérarchique fonctionnelle ;
- Il propose des étapes clés (documentation, revues) qui entraîne un bon suivi du projet ;
- Permet de garantir une certaine qualité (plan assurance qualité) ;
- Adapté à de grands projets ;
- Il est un modèle éprouvé car reproduit sur la production industrielle classique.

Nous avons présenté les modèles de cycle de vie d'un logiciel dans la section ci-dessus par le biais de les exposer, de les comparer et avec une justification de choix du modèle.

Mais dans la section suivante, nous présentons le choix des technologies et méthodes de conception utilisée dans la réalisation du projet.

4.2. Méthodologie de conception et technologies utilisées

Pour faire face à la complexité croissante des systèmes d'information, de nouvelles méthodes et outils ont été créées. La principale avancée des quinze dernières années jusqu'aux années 2005 réside dans la programmation orientée objet.

Face à ce nouveau mode de programmation, les méthodes de modélisation classique (telle MERISE) ont rapidement montré certaines limites et ont dû s'adapter. De très nombreuses méthodes ont également vu le jour comme Booch, OMT (Object Modelling Technique)...

Dans ce contexte et devant l'affluence de nouvelles méthodes de conception « orientée objet », l'Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'est apparu UML, qui est issu de la fusion des méthodes Booch, OMT et OOSE (Object Oriented Software Engineering).

Issu du terrain et fruit d'un travail d'experts reconnus, UML est le résultat d'un large consensus. De très nombreux acteurs industriels de renom ont adopté UML et participent à son développement. En l'espace d'une poignée d'années seulement, UML est devenu un standard incontournable.

Alors, cette section présente le choix des technologies et méthodes de conception utilisée dans la réalisation du projet de notre futur système.

4.3 Langage de modélisation utilisée

4.3.1. L'aspect choisi

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de la réaliser.

Il est nécessaire que nous choisissons UML comme langage de modélisation puisque nous allons utiliser le concept de l'orienter objet pour développer cette application.

UML est plus expressif, plus propre et plus uniforme que Booch, OMT, OOSE, et les autres méthodes. Cela signifie qu'il y a un bénéfice à passer à UML, parce qu'elle permettra aux projets de modéliser des choses qui n'auraient pas pu l'être avant. Les utilisateurs de la plupart des autres méthodes et langages de modélisation auront avantage à utiliser UML, puisqu'elle supprime toutes les différences non nécessaires de notation et de terminologie qui obscurcissent les similarités de bases de ces différentes approches. Ainsi, la méthodologie de

conception adoptée se base sur le choix de diagrammes UML adéquats. Nous avons utilisé cinq diagrammes : diagramme de cas d'utilisation, diagramme d'activités, diagramme de séquence, diagramme de classes, diagramme de déploiement.

4.3.2. L'UML

4.3.2.1. Historique

Né de la fusion des 3 méthodes objet dominantes (OMT, Booch et OOSE), puis normalisé par l'OMG en 1997, UML est rapidement devenu un standard incontournable [Rumbaugh J., Jacobson I. et Booch G., 1999]. UML n'est pas à l'origine des concepts objet, mais il en donne une définition plus formelle et apporte la dimension méthodologique qui faisait défaut à l'approche objet. Le but de cette présentation n'est pas de faire l'apologie d'UML, ni de restreindre UML à sa notation graphique, car le véritable intérêt d'UML est ailleurs ! En effet, maîtriser la notation graphique d'UML n'est pas une fin en soi. Ce qui est primordial, c'est d'utiliser les concepts objet à bon escient et d'appliquer la démarche d'analyse correspondante.

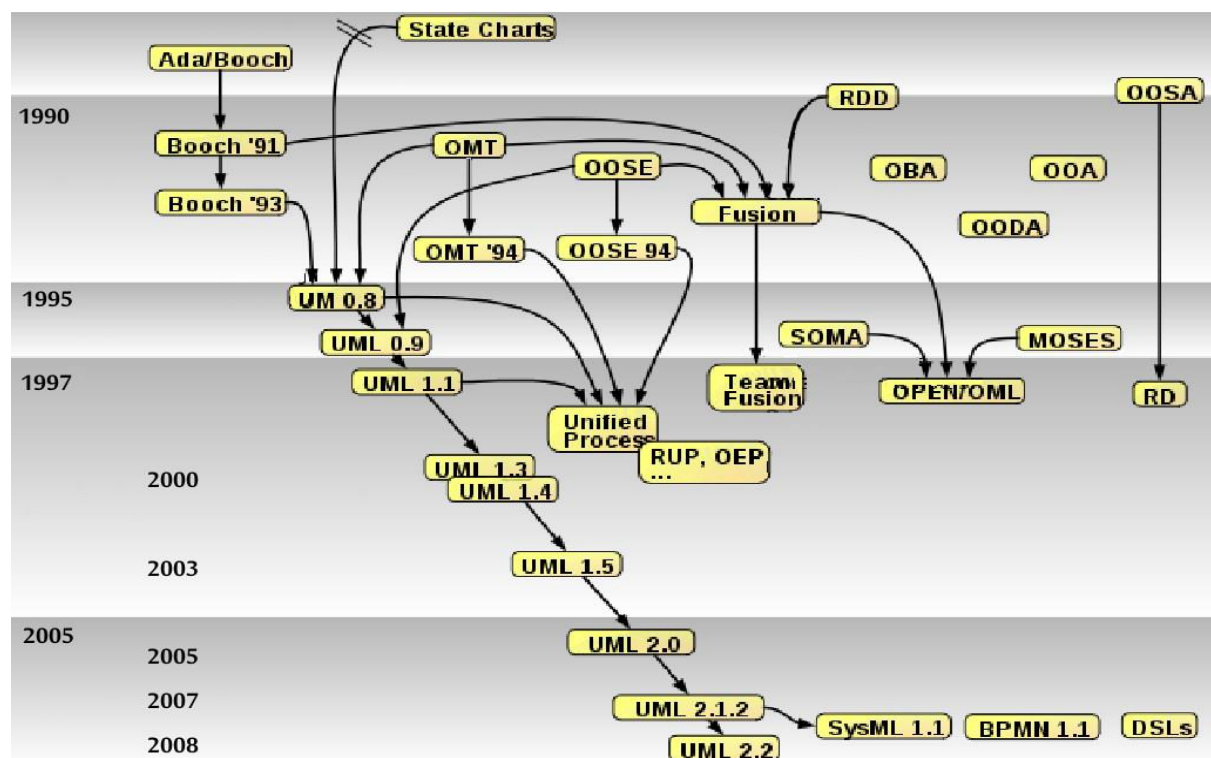


Figure 4.7: Evolution de l'UML

4.3.2.2. Définitions

L'UML traduit « langage de modélisation objet unifié » est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 80 [Rumbaugh J., Jacobson I. et Booch G., 1999].

- **Object Modeling Technique(OMT)** : a été développée par **James RUMBAUGH** dans le centre de Recherche et Développement de la société Générale Electric à la fin des années 80. Cette méthode connaît un certain succès en France et en Europe. OMT propose une triple perception du système d'information : une dimension statique qui décrit le schéma objet du système, une dimension dynamique qui décrit le changement d'états des objets en fonction d'événements, une dimension fonctionnelle qui décrit les processus de transformation des informations.

- **Object Oriented Development (OOD)** : Pionnier de l'orienté objet, Grady BOOCH publie en 1981 un article intitulé « Objet Oriented Development » qui présente une méthode de modélisation objet. Cette méthode applique le modèle objet pour synthétiser la structure logique, physique, statique et dynamique d'une application. Pour cela, elle utilise une notion expressive et clairement définie permettant de capturer des points de vue différents. La méthode Booch2 propose des diagrammes de classes, diagrammes d'instancié et diagrammes d'état transitions. Le niveau physique contient les diagrammes de modules et diagramme de processus.

- **Object Oriented Software Engineering(OOSE)** : la méthode OOSE de Ivar JACOBSON couvre tout le cycle de développement du logiciel, créée dans un centre de recherche d'Ericson (Suède), Elle repose essentiellement sur l'analyse des besoins des utilisateurs à l'aide des diagrammes de cas d'utilisation formalisés par Ivar JACOBSON dans sa méthode puis introduits dans UML qui décrit, sous la forme d'actions et de réactions, le comportement d'un système du point de vue de l'utilisateur, définissent les limites du système et les relations entre le système et l'environnement. Ils permettent d'introduire une manière spécifique d'utiliser un système. C'est l'image d'une fonctionnalité du système, déclenchée en réponse à la stimulation d'un acteur externe.

4.3.2.3. Les caractéristiques de ses diagrammes

Pour un système donné, les diagrammes de conception de l'UML doit satisfaire les caractéristiques suivants [Laurent A., 2010]:

- Combinés, les différents types de diagramme UML offrent une vue complète des aspects statiques et dynamiques d'un système ;
- Par extension et abus de langage, un diagramme UML est aussi un modèle ;
- UML permet de définir et de visualiser un modèle, à l'aide de diagrammes ;
- Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle, pas « le modèle » ;

- Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) ;
- Un type de diagramme UML véhicule une sémantique précise (un type de diagramme offre toujours la même vue d'un système).

4.3.2.4. Ses Objectifs

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. Il a été conçu comme un ensemble de notation qui vise à être utilisées dans différentes étape de la modélisation objet, de l'expression des besoins jusqu'à l'implantation. Il s'agit de la notation synthétique et intuitive et accessible par différentes intervenants de la modélisation.

UML est donc un langage de modélisation avec plusieurs objectifs qui font un véritable outil de communication [Laurent A., 2010]:

- Comprendre et décrire les besoins ;
- Spécifier un système ;
- Établir l'architecture logicielle.

4.3.2.5. UML face à la programmation

Pour programmer une application, il ne convient pas de se lancer tête baissée dans l'écriture du code, mais il faut organiser ses idées, les documenter puis organiser la réalisation en définissant les modules et les étapes. La version d'UML est caractérisée par le nombre de diagrammes qui contient [Muller P.-A., 1998].

4.3.2.6. Les points forts et points faibles d'UML

Le tableau 4.1 suivant présente les avantages et les inconvénients d'UML.

Tableau 4.1: Les avantages et les inconvénients d'UML

Points forts	Faiblesses
UML est un langage formel et normalisé : <ul style="list-style-type: none"> • Permet une grande précision, • Assure la pérennité, 	UML nécessite un apprentissage et de l'expérience : <ul style="list-style-type: none"> • UML n'est pas à l'origine de

<ul style="list-style-type: none"> • Facilite l'utilisation d'outils <p>UML est un outil qui facilite la communication :</p> <ul style="list-style-type: none"> • Propose un cadre d'analyse, • Permet la représentation d'éléments abstraits et complexes, • Offre une grande expressivité grâce à sa polyvalence et à sa souplesse. 	<p>l'approche objet mais en précise les concepts sous-jacents,</p> <p>UML n'est pas une méthode (et n'en propose pas) :</p> <ul style="list-style-type: none"> • Cela va probablement changer ultérieurement, • Certaines méthodes intègrent au moins partiellement l'utilisation d'UML
UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce à ces différents diagrammes ;	UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et donne une définition plus formelle.
Les diagrammes d'UML permettant de spécifier, de construire, de documenter, de visualiser et de manipuler les systèmes informatiques et ils permettent d'isoler certaines parties des modèles pour les rendre plus aisément compréhensibles.	

4.3.2.7. Les grandes lignes essentielles de l'UML

L'UML, « Langage de modélisation d'objet unifié », a pour but de clarifier et de simplifier la conception et la représentation d'un système donné et de son fonctionnement. Particulièrement, UML est un langage de choix, la schématisation simple compréhensible par l'homme et la machine de ses étapes essentiels [Hajalalaina A. R., 2016].

Comme étape de modélisation d'UML, les différents diagrammes persistent. Les étapes que la modélisation contient distinguent en deux types de vues:

- Les vues statiques ;
- Les vues dynamiques.

Les vues statiques représentent le système physiquement qui comprend les diagrammes suivants :

- *diagrammes d'objets*, il sert à représenter les instances de classes (objets) utilisées dans le système ;

- *diagrammes de classes*, il représente les classes intervenant dans le système ;
- *diagrammes de cas d'utilisation*, il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système ;
- *diagrammes de composants*, il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...) ;
- *diagrammes de déploiement*, utilisés pour représenter l'architecture physique d'un système en utilisant des nœuds et des connexions entre ces nœuds ;

Les vues dynamiques montrent le fonctionnement du système comprends les diagrammes suivants :

- *diagrammes de séquence*, représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs ;
- *diagrammes de collaboration* ; Scénario d'un cas d'utilisation: activités des objets et des messages échangés ;
- *diagrammes d'état transitions* permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants ;
- *diagrammes d'activités* permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

Tableau 4.2 : Classement des diagrammes

CLASSIFICATION	DIAGRAMME
Besoins des utilisateurs	– Diagramme des cas d'utilisation
Structure statique	– Diagramme de Classe – Diagramme d'objet
Dynamique des objets	– Diagramme états-transition – Diagramme d'activités
Interactions entre objets	– Diagramme de séquence – Diagramme de collaboration
Réalisation et déploiement	– Diagramme de composants – Diagramme de déploiement

Les différents diagrammes UML qui sont les diagrammes structurels et comportementaux sont représentés par la figure 4.8 suivante [RAZAFINDRAGY J.A., 2016].

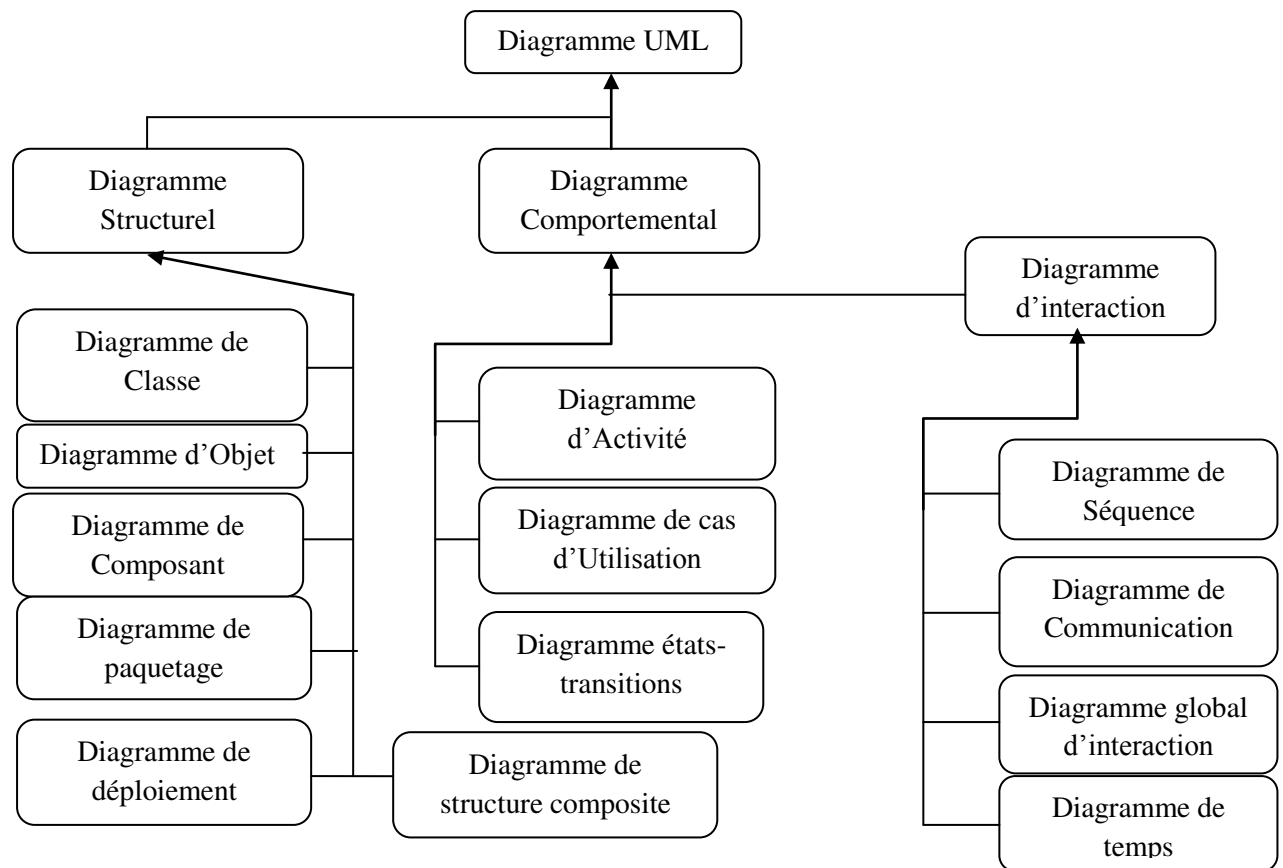


Figure 4.8 : Les différents diagrammes UML [adapté aux différents diagrammes de

a. Diag

l'UML de RAZAFINDRAGY A.J.A., 2016]

Les diagrammes de cas d'utilisation permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Dans notre cas nous allons utiliser deux types de diagrammes : le diagramme préliminaire et les diagrammes détaillés. Le diagramme de cas d'utilisation préliminaire permet de représenter les principaux cas d'utilisation du système, ces cas d'utilisation seront ensuite détaillés dans les Cas d'utilisation détaillés. Le diagramme de cas d'utilisation détaillés permet donc de détailler les cas d'utilisation du cas d'utilisation préliminaire, ces principaux cas d'utilisation seront traités sous forme de processus [Laurent A., 2010].

- **Acteur**

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

- **Cas d'utilisation**

C'est un graphe dont les nœuds sont des acteurs ou des cas d'utilisation et les arcs sont des relations de communication entre acteurs et cas d'utilisation ou des relations standardisées entre cas d'utilisation.

- **Les relations entre cas d'utilisation**

UML définit deux types de relations standardisées entre cas d'utilisation détaillées ci-après :

- Une relation d'inclusion, formalisée par le mot-clé « include » ;
- Une relation d'extension, formalisée par le mot-clé « extend ».

- **Représentation graphique**

La figure 4.9 représente le formalisme du diagramme de cas d'utilisation.

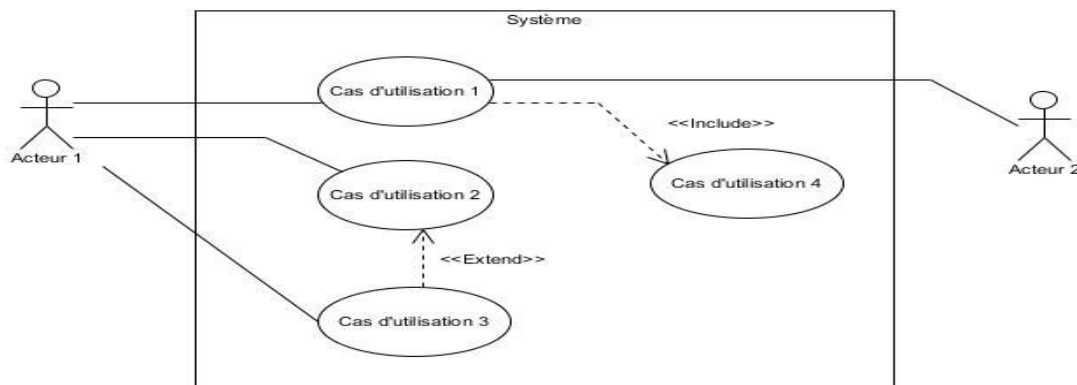


Figure 4.9 : Formalisme du diagramme de cas d'utilisation.

b. Diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur ; il indique les objets que l'acteur va manipuler, et les opérations qui font passer d'un objet à l'autre [Laurent Audibert, 2010].

Il permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme de cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par

le biais de messages. La représentation se concentre sur l'expression des interactions et met l'accent sur la chronologie des envois de messages.

- **La ligne de vie**

La ligne de vie représente un participant à une interaction objet ou acteur.

- **Le message**

Le message définit la communication particulière entre des lignes de vie objets ou acteurs. Différents types de message (ou actions) peuvent transiter entre les acteurs et objets :

- **message simple** : Message qui ne spécifie aucune caractéristique d'envoi ou de réception particulière ;
- **message minuté** (timeout) ou **message avec durée de vie** : Il bloque l'expéditeur pendant un temps donné en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié ;
- **message synchrone** : Il bloque l'expéditeur jusqu'à la prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur à la prise en compte du message c'est-à-dire l'émetteur devient passif et le récepteur actif ;
- **message asynchrone** : Il n'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré ;
- **message dérobant** : Il n'interrompt pas l'exécution de l'expéditeur et le message est mis en attente dans une liste d'attente de traitement chez le récepteur.

- **Représentation graphique**

La figure 4.10 représente le formalisme de diagramme de séquences type.

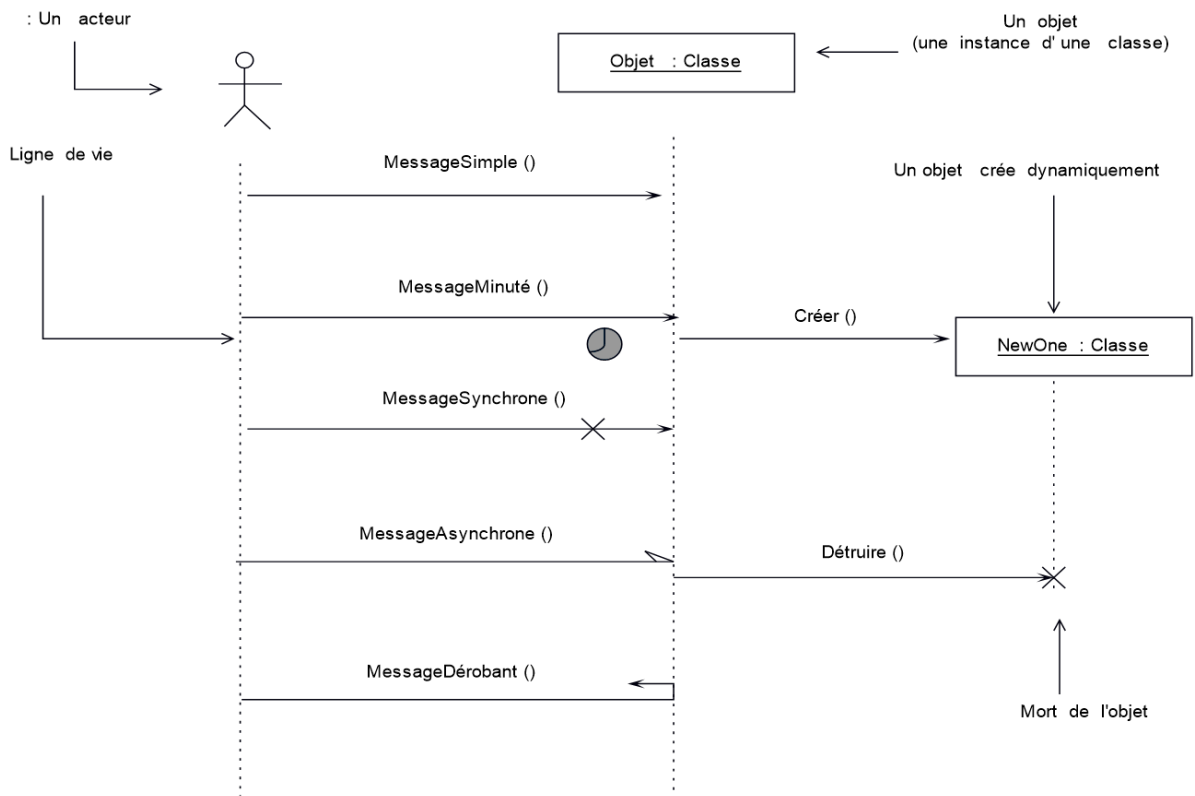


Figure 4.10 : Formalisme du diagramme de séquence.

c. Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Les diagrammes de classe appropriés représentent les classes participant à la réalisation d'un cas d'utilisation et la relation entre elle. Les outils de cette notion peuvent gérer des interfaces graphiques pour le schéma de la conception, ils peuvent aussi établir les classes essentielles de l'application et enfin, ils peuvent générer automatiquement les codes écrits en langage spécifié par l'utilisateur à partir des classes déjà construites [Laurent Audibert, 2010].

- **Définition des éléments**

Les éléments d'un diagramme des Classes sont les classes et les relations qui les lient. Nous allons expliquer ci-dessous chacun de ces éléments :

Les **classes** sont les modules de base de la programmation orientée objet. Une classe est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe. La section centrale définit les propriétés de la classe et la section du bas énumère les méthodes de la classe.

Une **association** est une relation générique entre deux classes. Elle est modélisée par une ligne reliant les deux classes. Cette ligne peut être qualifiée avec le type de relation, et peut également comporter des règles de multiplicité (par exemple un à un, un à plusieurs et plusieurs à plusieurs) pour la relation.

Une **relation de composition** est indiquée par une ligne avec un "diamant" rempli. Si une classe ne peut pas exister par elle-même, mais doit être un membre d'une autre classe, alors elle possède une relation de composition avec la classe contenant.

Une **relation de dépendance** est représentée par une flèche pointillée. Quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe.

Les **agrégations** indiquent une relation de contenant-contenu. Elle décrite par une relation "possède". Une relation d'agrégation est représentée par une ligne avec un "diamant" creux.

Une **relation de généralisation** est l'équivalent d'une relation d'héritage en terme orientés objet (relation "est-un"). Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe "parent".

- **Représentation graphique**

C'est le formalisme du diagramme de classe qui est représenté par la figure 4.11:

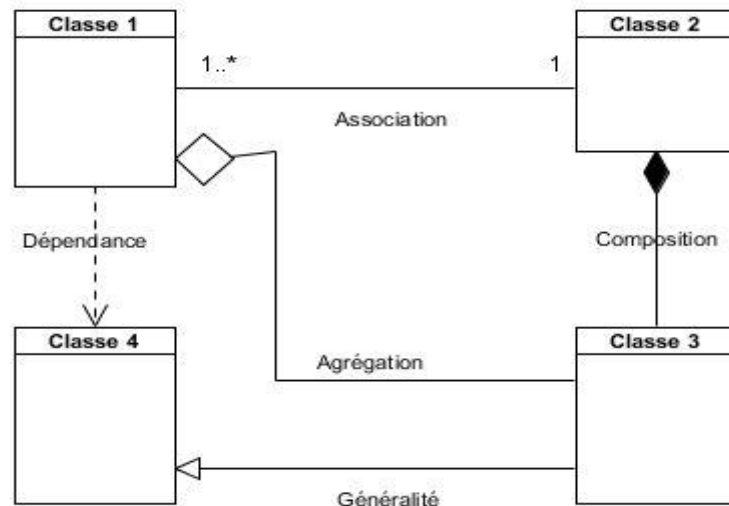


Figure 4.11 : Formalisme de diagramme de classe

- **Cardinalités**

La cardinalité précise le nombre d'instances qui participent à une relation.

L'expression des cardinalités d'une relation en UML :

- **n** : exactement "n" (n, entier naturel > 0) ;
- **n..m** : de "n" à "m" (entiers naturels ou variables, m > n) ;
- ***** : plusieurs (équivalent à "0..n" et "0..*") ;
- **n..*** : "n" ou plus (n, entier naturel ou variable).

d. Diagramme d'activités

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Nous pouvons rattacher un diagramme d'activités à n'importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément. Dans ce projet les diagrammes d'activités sont attachés aux diagrammes de cas d'utilisation détaillés [Laurent Audibert, 2010].

- **Représentation graphique**

C'est le formalisme du diagramme d'activité qui est représenté par la figure 4.12:

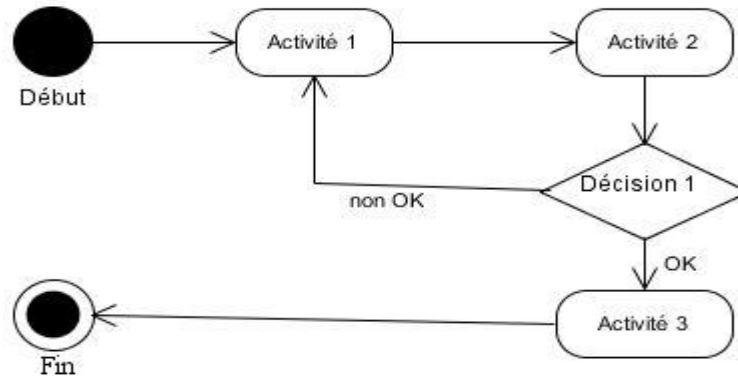


Figure 4.12 : Formalisme du diagramme d'activité

e. Diagramme de déploiement

Un diagramme de déploiement est un diagramme de classes ou un diagramme d'objets représentant les nœuds ou les instances de nœuds sur lesquels le système s'exécute. Il propose une vision statique de la topologie du matériel sur lequel s'exécute le système et il montre les associations (connexions) existant entre les nœuds du système.

Un nœud est un élément physique matériel sur lequel le système s'exécute. Il peut être un processeur, un périphérique ou un réseau [Laurent Audibert, 2010].

- **Représentation graphique**

C'est le formalisme du diagramme de déploiement qui est représenté par la figure 4.13 :

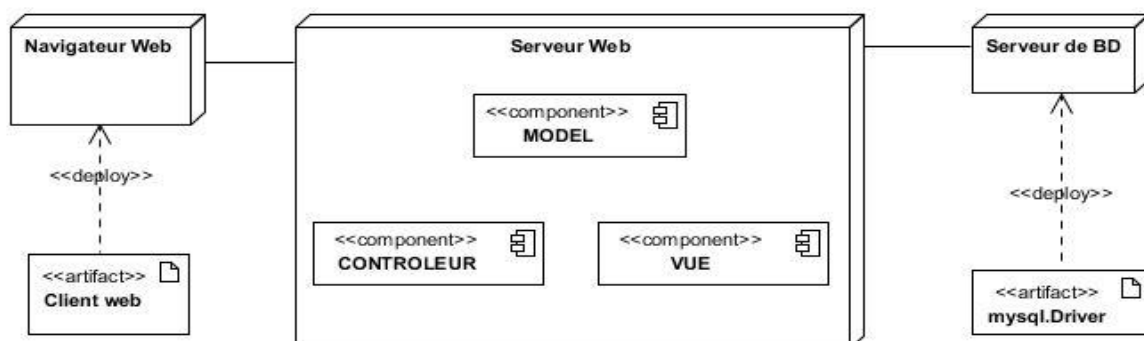


Figure 4.13: Formalisme du diagramme de déploiement

4.2.3.7. Choix des outils pour la modélisation en UML

– Les outils

Il y a plusieurs outils avec lesquels on peut faire les diagrammes d'UML, ce sont :

- Win' Design ;
- Visual Paradigm ;
- UML Designer ;
- BO UML ;
- Magic Draw.

– Comparaison des outils

Chaque logiciel sera évalué selon sa qualité à traduire différents éléments d'un diagramme de classes.

Le tableau 4.3 nous montre la comparaison des quelques outils de conception les plus utilisées [Wikipédia, en ligne 2016].

Tableau 4.3 : Comparaison des outils de conception d'UML

Outils	Fiabilité	Efficacité	Open Source	Coût	Plateforme/ SE
Win' Design	Oui	Oui	Non	Payante	Windows
BO UML	Oui	Oui	<5.0 Oui > 5.0 Non	<5.0 Gratuit >5.0 Payante	Linux, Windows et Mac OS X
Visual Paradigm for UML	Oui	Oui	Non	Payante	Multiplateforme
Magic Draw	Oui	Oui	Non	Payante	Multiplateforme
UML Designer	Oui	Oui	Oui	Payante	Multiplateforme

Dans la conception de notre projet, nous choisissons BO UML pour l'outil de modélisation.

Chapitre 5 : Analyse détaillée du projet web

Après avoir présenté les méthodes et les notations utilisées de notre projet, le présent chapitre nous permet d'analyser les détails du projet de notre futur système pour chaque étude, et ceci en recensant l'étude préalable, l'étude de l'existant, l'analyse des besoins, étude de faisabilité, type et l'objectifs généraux de l'application, les critiques et propositions des solutions.

5.1. Étude préalable

En réalité, tout projet doit passer par une phase d'analyse et de conception. Cette phase s'effectue au moyen d'entretiens avec les différents acteurs du projet ainsi que l'analyse des documents existants. Cela permet d'éprouver les détails du domaine à étudier et ensuite apporter les solutions fonctionnelles et techniques.

5.2. Etude de l'existant

L'étude de l'existant consiste à collecter des données indispensables à la création de l'application, il consiste aussi à déterminer si le projet est faisable ou non, c'est-à-dire que si la création de cette application apportera donc du bénéfice pour la société.

5.2.1 Analyse de l'existant

L'analyse permet de considérer les différents problèmes et les besoins sur l'administration de vente de l'accueil du parc national Isalo. L'objectif de l'analyse est d'obtenir les informations concernant les tâches à exécuter sur la vente et l'enregistrement des données dans la base pour un but de l'améliorer. Elle consiste à réaliser de manière objective les buts à atteindre tout en considérant les contraintes aux nouvelles solutions.

L'interview nous a permis d'éprouver les objectifs générales et les essentiels problèmes de la gestion de vente à l'accueil du parc comme :

- Traitement de la gestion sous tableur Excel ;
- En remplissant le facture et le fait de faire le calcul sont plutôt faible ;
- Travail à refaire sur papier pour remplir la note de débit ou facture qui prend beaucoup de temps;
- Exécution des tâches allongées et avec l'incertitude de calcul ou d'avoir oublié quelques une des informations;
- En remplissant le PCA ou la recette DEAP qui est journalière, l'hôtesse a besoin d'être calme pour avoir la surette de sa calcul ou bien le refaire.
- Recherche des donnés par triage sur le tableur.

5.2.1.1 Matériel existant

Le MNP PNI est doté d'avoir de bon nombre de matériels informatiques. Leurs caractéristiques et leur liste sont présentés dans le tableau.

Tableau 5.1: Liste des matériels informatique et leurs caractéristiques

	Caractéristiques	Nombres	Emplacement
Ordinateur	FUJITSU Siemens Core i5 RAM : 4Go. Disque Dur SATA 500 Go Lecteur DVD, Ecran LED 25 Pouces	1	DP
		2	CVSR, CVADEE
		1	CV Ecot
		1	CVAF
	NEC COMPUTER SAS Pentium 4 RAM : 1Go Disque Dur IDE 250 Go Lecteur DVD, Ecran LCD 22 Pouces	1	Accueil
		1	Logisticien
	HP Probook 450 G2 Core i5 RAM : 4Go Disque Dur SATA 500 Go Lecteur DVD, Ecran LED 25 Pouces	1	CVAF
Imprimante	Canon HP Deskjet	2	CVSR
		1	Comptable
	XEROX Laser M118	1	Logisticien

5.2.1.2 Logiciel existant

Le MNP PNI a des différents logiciels pour son réalisation des travaux. Le tableau 5.2 suivant comporte la liste de ses diversités logicielles.

Tableau 5.2 : Liste de divers logicielles existant.

	Nom	Version
Système d'exploitation	Windows XP Professionnel 32 Bits	
	Windows 7 Edition Familial Premium	
	64 bits	

	Windows 7 Professionnel 64 bits Windows 8 Professionnel 64 bits	
Autres logiciels	Office	2007, 2010, 2013
	Arc Gis	8.0
	Mozilla Firefox	41.0.2
	Antivirus Avast	8.0
	Smadav 2015	41.2

5.2.1.3 Mode de gestion de vente

La gestion de vente au service d'accueil du parc national Isalo, plus particulièrement la vente de tickets, s'effectue en mode physique. Même s'il y a de tableur Excel pour la gestion, elles ont utilisé le papier pour un support physique et pour la note de débit car quelqu'un a pu changer dans les données dans la base, à part les deux hôtesse et leur chef de volet, changée les informations stockées dans leur base de données à cause de la manque de sécurisation de données.

En faisant le calcul, en présence des bon nombres de clients et de bruits, il est fortement possible que les hôtesse ont fait des erreurs pendant le compte. Si elles ont un peu de temps libre, ce qui veut dire pas de client, elles ont commencés d'enregistrer les données dans le tableur et dans le support physique et refaire le calcul de toutes ventes effectuées de la journée.

5.2.2. Etude de faisabilité

L'étude de faisabilité marque l'initialisation du projet en fonction des ressources de l'établissement. Sur le côté organisationnelle et technique, pour pouvoir créer une application sans risque, les matériels nécessaires pour effectuer la conception et la réalisation du projet sont :

- Intel Pentium IV de fréquence supérieure ou égale à 2 GHz ;
- RAM : au moins 512Mo ;
- Disque dur : à partir de 80Go ;
- Ecran à partir de 15'' ;

Après l'étude de faisabilité organisationnelle et technique, nous allons passer maintenant à l'étude de faisabilité économique. Du point de vue logiciel à utiliser pour la réalisation de

l'application, on constate que le Service d'accueil dispose toutes les ressources nécessaires. Alors, passons directement aux critiques de l'existant.

5.2.3. Critique de l'existant

Des renseignements et des activités sont répétitifs et à fonction exponentiel, les problèmes les ont aussi. En plus, le responsable d'accueil a fait une double corvée et avec de problème de délayage à l'information qui les concernent, cela a causé une perte de temps pour lui. Chacun des chefs de volets ont besoins des informations relatif aux activités et aux clients, alors que c'est aussi l'une des problèmes pour les fournir ces informations car cela ont besoins de temps pour les classer dans un même genre. Malgré l'utilisation d'un ordinateur pour le traitement des données, le travail est aussi fait manuellement et rencontre une difficulté pour la recherche des informations. Il n'y a pas encore de système qui traite automatiquement les calculs de montant total d'une note de débit.

5.2.3.1 La critique positive

Même s'il y a des problèmes entre leur traitement de données, le travail a bien fonctionné et répond à leur besoin nécessaire grâce au tableur Excel. En plus, ils ont la même mode de travail pour l'inter change des données en utilisant les supports comme la clef USB qui rend plus agile les responsables de chaque volets en faisant en sorte que les donnees soient confronté afin d'avoir vite fait les rapports à la fin de mois pour les envoyer au Siege à Antananarivo. Ils ont les résultats satisfaisants.

5.2.3.2 La critique négative

A notre époque d'aujourd'hui, tous semblent être informatisé sans parler des travaux qui doivent faits en laborant ou d'autre. Alors qu'au service d'accueil, il y a un ordinateur pour faire automatisé l'œuvre mais il ne l'utilise que pour stocker les données seulement et rien de plus. Et pour gérer ces derniers, il fait appel à un autre ordinateur pour pouvoir élaborer les statistiques car l'hôtesse rencontre un petit problème de manipulation et seul leur chef de volet qui peut les régenter.

5.2.4. Solutions proposées

Dans le cadre de la résolution des problèmes cités ci-dessus, une application web va améliorer le travail et l'administration de vente au service d'accueil au sein du MNP PNI. Tout en utilisant le PHP, l'architecture MVC, et un bon Framework nous arriverons à faire gagner le temps à l'utilisateur de cette interface et lui permettre de faire un saut vers l'évolution du travail manuel fatigant au travail informatisé qui va comporter ce qui suit de:

- Créer une application capable de stocker et de faire une recherche aux besoins pour les données existant ;
- Pour avoir plus de performance, il vaut mieux suivre la technologie informatique moderne, c'est-à-dire il faut développer une application web avec les nouvelles technologies web, en pratiquant la norme IHM ;
- De nos jours, il y a une solution qui peut gérer l'ensemble de tout un travail d'une entreprise (en particulièrement la gestion commercial, la gestion comptable, la gestion de stock...) en utilisant un logiciel visant à unifier leur système d'informations de chaque branche. Il vaut mieux de faire en usage ce logiciel comme le Ciel Compta et le logiciel SOARI. Mais le problème de ce logiciel est, non seulement son cout de la licence de progiciel de gestion intégré coûte très chère mais en plus sa mise en place doit être faite par des professionnels et tous le système d'informations doit être revu ;
- Il est possible aussi d'importer la base de données en Excel dans une application web. PHP peut gérer les données en Excel pour les autres informations indispensables afin que les données restent comme ils sont pour les autres modules qui ne sachent qu'utiliser le tableur Excel.

5.2.5 Solution retenue

Parmi ces solutions possibles, après la conversation que nous avons faite avec le responsable, il a décidé de créer une application pour administrer leur base de données et leur vente au service d'accueil et pour automatiser la tâche et de faire le maximum possible de sécurité. Donc, notre pensée est orientée à la création d'une application avec le technique moderne de la technologie web sur nous-même et qui peut aussi importer une base en Excel.

5.3 Type de l'application

En générale, c'est une application web pour faciliter les services d'Accueil du Parc National Isalo et suivant la technologie moderne avec une connexion avec le téléphone mobile pour communiquer.

5.3.1. Objectifs généraux de l'application

Dans ce projet, nous voulons crée une application web dynamique.

Les grandes lignes de cette application sont donc de :

- faire l'optimisation du référencement c'est-à-dire gérer les sessions, puis, progresser les temps de chargement des pages ;
- faire couramment le calcul et le triage des données ;

- produire rapidement le statistique ;
- faire fonctionner le transfert de message avec un téléphone mobile.

5.3.2. Rédaction du plan projet

Le plan du projet est une assurance pour l'accomplissement d'une application.

5.3.3. Planning du projet web

Le planning est un des éléments clés du plan projet car il engage la responsabilité de l'équipe de réalisation sur les dates de livraisons. L'application web dynamique que nous allons créer doit être livrable à la fin du mois de Mars 2017. Pour cela, les trois mois de stage suivent le planning suivant.

Tableau 5.3 : Planning du projet

Décembre/Mi-janvier	Mi-janvier/fin-janvier	Février/Mars
Observation de tous les services actifs et leur déroulement.	Exploitations des données	Réalisation de l'application
Sondages des archives retenant les contenus de l'application.	Analyse et conception de l'application	Essai et Utilisation.
Recueil des informations et des données sur le déroulement de collecte des informations.		
Répartitions des équipes.		

5.3.4. Constitution de l'équipe

Notre responsabilité est de créer une application pour notre école aussi. L'équipe du projet se font en 5 groupes différents (s'il existe). Ces 5 groupes sont divisés dans le tableau suivant selon leurs actions :

Tableau 5.4: Liste des équipes du projet et leurs actions

Groupe	Actions
Concepteur et webdesigner	Il conçoit l'architecture interactionnelle, l'arborescence et la navigation de l'application en respectant les indications graphique.
Ergonome	L'ergonome collecte toutes les informations sur le

	comportement possible des futurs internautes de l'application web. Il travaille profondément avec le webdesigner pour définir le type de navigation.
Infographe	Formé aux principaux outils du marché (flash, Dreamweaver, Photoshop, Image Read), l'infographiste doit suivre les consignes du Directeur artistique et réaliser les éléments graphiques, ainsi que les Template des pages.
Rédacteur	Il s'agit du responsable des contenus, de leur agencement et du style d'expression. Le rédacteur doit structurer et adapter pour le web le contenu informationnel fourni par le client.
Développeur	En fonction du besoin, l'équipe de développement réalisera de simples intégrations «statique » (découpage et intégration HTML /JavaScript) ou de véritables développements dynamiques (modélisation de base de données, interaction avec l'application).

Chapitre 6 : Conception du projet

La gestion de projet et l'utilisation de la notation UML est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Ce présent chapitre nous a permis de présenter la conception du notre projet par la bais du dictionnaire des données et des différents diagrammes.

6.1 Règles de Gestion

La règle de gestion est une manière utilisée pendant une conception pour pouvoir éviter la rupture ou la redondance. Donc, notre application a ses règles de gestion qui sont :

- **RG1** : Le client est identifié par sa nationalité, le nom d'hôtel qui l'héberge pendant son séjour, son agence de voyage ou tours opérateur, son classe d'âge et le nombre total.
- **RG2** : Le client et le guide sont classés comme des visiteurs mais leur fonction les différencie en général.
- **RG3** : Le ticket d'entrée peut être une journée ou deux jours ou trois jours mais pour les quatre jours et plus c'est une combinaison de ticket.
- **RG4** : Le ticket vente Tana (VTA) ne doit pas être déchiré qu'à l'accueil du parc pour qu'il soit valide.
- **RG5** : Le client a la permission de visiter un ou plusieurs circuits.
- **RG6** : Un client doit payer un et un seul facture ou note de débit.
- **RG7** : Le prix de ticket dépend de la durée de circuit que le client souhaite.
- **RG8** : Un guide doit avoir au moins un numéro de téléphone.
- **RG9** : Les clients doivent être guidés par un ou plusieurs guides.
- **RG10** : Le circuit est composé de nom du site, de durée de visite et de la désignation que le client a pu visiter.
- **RG11** : Avant de donner le ticket au client, il faut mentionner sur le ticket tous les informations collectées avec le client comme leur agence de voyage, l'hôtel...

6.2 Dictionnaire de données

Le dictionnaire de données recense toutes les données qui pourront être manipulées durant la conception et peuvent être dans la réalisation.

Cette étape aboutit au dictionnaire de données qui devra comporter ni synonyme ni données calculée. Et pour cette application, on les montre dans le tableau 6.1 :

Tableau 6.1 : Dictionnaire de données de notre application

Rubrique	Définition	Type	Longueur	Etat	Format
AGV	Agence de voyage qui transporte le client	AN	30	E	
Classe_Age	Classe d'âge pour qualifier le prix de ticket	AN	8	E	
Connaissance	La connaissance linguistique d'un guide	A	20	E	
Date_Achat	La date que le client fait l'achat du ticket d'entrée	D	10	E	dd/mm/yyyy
Date_Visit	La date que le client fait la visite du parc	D	10	E	dd/mm/yyyy
Designation	Désignation du circuit qui peut visiter	AN	10	E	2CPNNOBC RT
Duree	La durée de visite dans le circuit	A	10	E	
Nationalite	La nationalité du client	A	15	E	
NumGuide	le numéro de rang d'un guide	N	3	E	
Montant	La totalité de prix de tickets qui sont achetés par le client	N	8	E	1 000 000
Nom_Guide	Le nom de guide	AN	50	E	
Nombre	Le nombre du jour par groupe	N	2		
Nombre_de_journee	Le nombre de jour qui spécifie la durée de séjours du client	AN	15	E	
Hotel	Le nom de l'hôtel que le client héberge à Ranohira	AN	10	E	H1
Nom_Circuit	Le nom du site d'un circuit	AN	20	E	
Num_Deb	Le numéro de début du ticket	N	5	E	18364
Num_Fin	Le numéro de la fin du ticket	N	5	E	18382
Num_Telef	Le numéro téléphone de guide	AN	15	E	034 95 916 92
Prenom	Prénom(s) d'un guide	AN	50	E	
PU	Le prix unitaire de ticket	N	5	E	200000

Type	Le type de ticket	AN	4	E	AE1
------	-------------------	----	---	---	-----

Légende : AN : Alphanumérique, A : Alphabétique, N : Numérique, E : Elémentaire.

6.3. Diagramme de cas d'utilisation

6.3.1. Identification des acteurs

Les acteurs du système sont tous des acteurs humains, c'est-à-dire qu'il n'y a pas interaction du système a d'autre systèmes informatiques. Ces acteurs sont :

- Les Hôtesse qui utilisent l'application en faisant l'enregistrement des données et les calculs automatique des recettes du jour.
- L'administrateur du système qui est le Chef de Volet Ecotourisme du PNI et gère tous les résultats de la vente et les données d'Ecotourisme.

6.3.2. Identification des cas d'utilisation

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes intentions « métier » selon lesquelles il utilise le système.

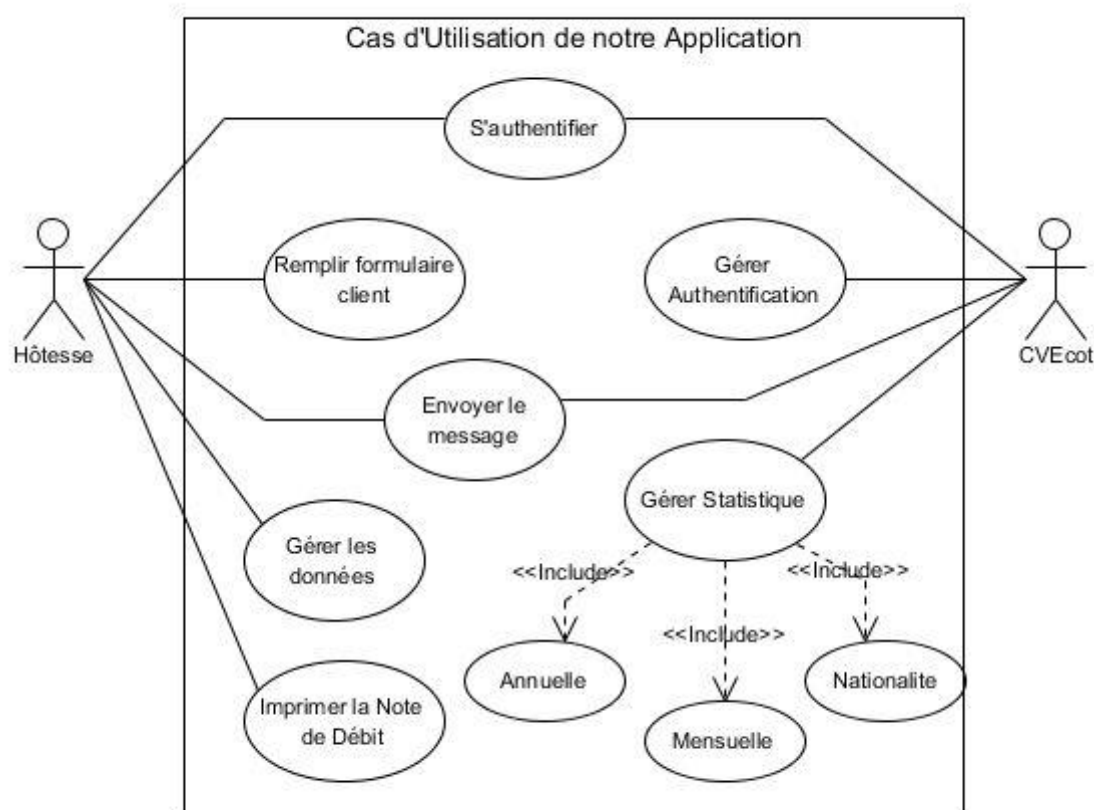


Figure 6.1 : Diagramme de cas d'utilisation de l'Application

6.3.3. Spécification détaillé des cas d'utilisation d'un système

Il modélise une interaction entre le système informatique à concevoir et l'utilisateur qui interagisse avec le système c'est-à-dire que les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. Il capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il partage la fonctionnalité du système en unités cohérentes, les cas d'utilisation : ayant un sens pour les acteurs. Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs.

Pour élaborer les cas d'utilisation, il faut se fonder sur des entretiens avec les utilisateurs. Un diagramme de cas d'utilisation est composé par : l'acteur et le Cas d'utilisation.

L'acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Le cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie.

Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service [Hajalalaina A. R., 2016].

Description du cas d'utilisation « S'Authentifier »

Tableau 6.2: Raffinement du cas d'utilisation «S'authentifier»

Sous cas d'utilisation:	S'authentifier
Acteur :	CV Ecot ou Hôtesse (Utilisateur)
Pré-condition:	Avoir login et mot de passe correct.
Post-condition:	CV Ecot ou Hôtesse authentifié.
Description de scénario de base :	<ol style="list-style-type: none"> 1. L'utilisateur saisit son login et son mot de passe. 2. L'utilisateur clique sur «Login ». 3. Le système vérifie la combinaison login et mot de passe. 4. Login et mot de passe correcte, le système affiche l'accueil.
Exception :	S'il n y a aucun Hôtesse enregistré dans la base, un message s'affiche « Pas d'hôtesse enregistré ».

En s'authentifiant pour avoir un accès au système, l'utilisateur parvient à son compte via le mot de passe et le nom d'utilisateur qu'il a choisi lors de la création de celui-ci. Ces données seront enregistrées dans la table intitulée utilisateur et c'est la raison pour laquelle le contrôleur effectue une opération de lecture sur cette table à la recherche de l'utilisateur en question avant de permettre l'accès ou le refuser.

6.4 Diagrammes de séquences

6.4.1 Diagramme de séquence de cas d'utilisation «S'Authentifier»

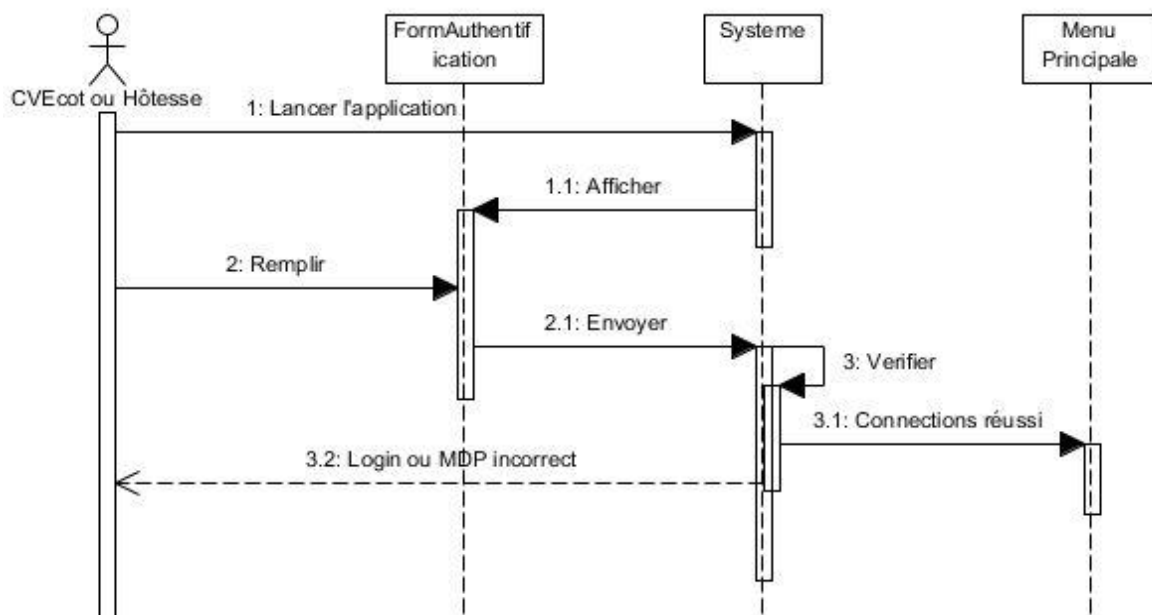


Figure 6.2 : Diagramme de séquence de l'Authentification

6.4.2 Diagramme de séquence de cas d'utilisation «Voir la statistique des données »

Le diagramme sur la figure 6.3 suivant présente le diagramme de séquence du cas d'utilisation « Voir la statistique des données ».

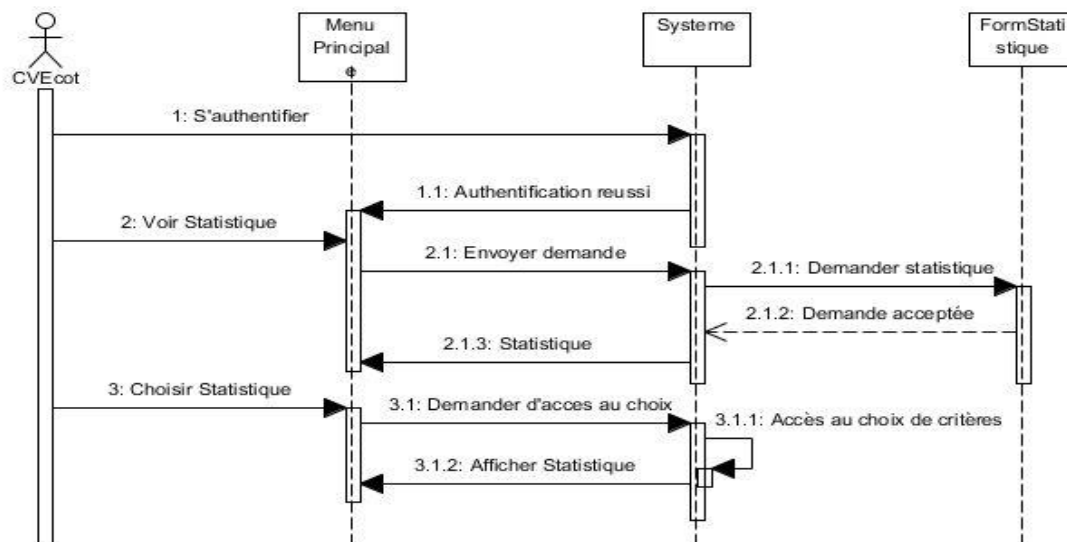


Figure 6.3: Diagramme de séquence du cas «Voir la statistique des données Ecotourisme».

6.4.3 Diagramme de séquence de cas d'utilisation «Calcul des données»

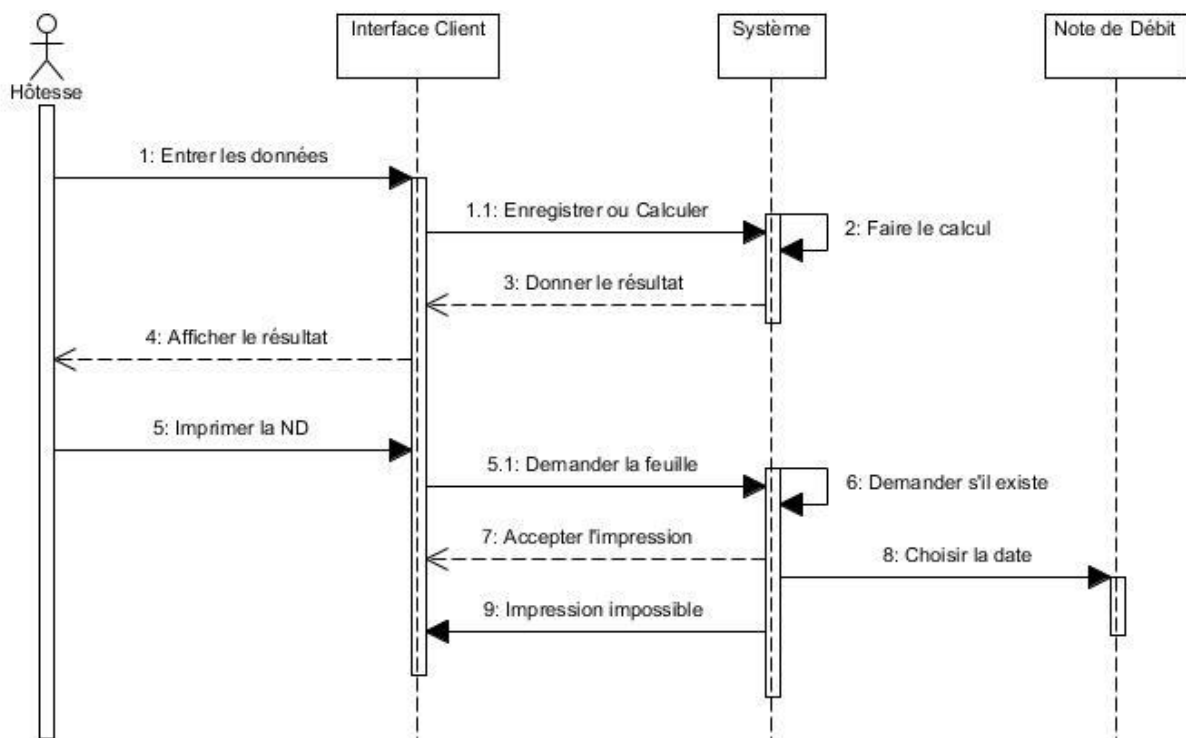


Figure 6.4 : Diagramme de séquence de cas d'utilisation « Calcul des données»

6.5 Diagramme d'activité

Le diagramme d'activité est attaché à une catégorie de classe et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "**flot de contrôle**". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les **conditions** de séquencement. Il pourra comporter des synchronisations pour représenter les déroulements parallèles.

La notion de couloir d'activité va décrire les responsabilités en répartissant les activités entre les différents acteurs opérationnels. Chaque activité sera placée dans une colonne (couloir) qui correspond à l'acteur.

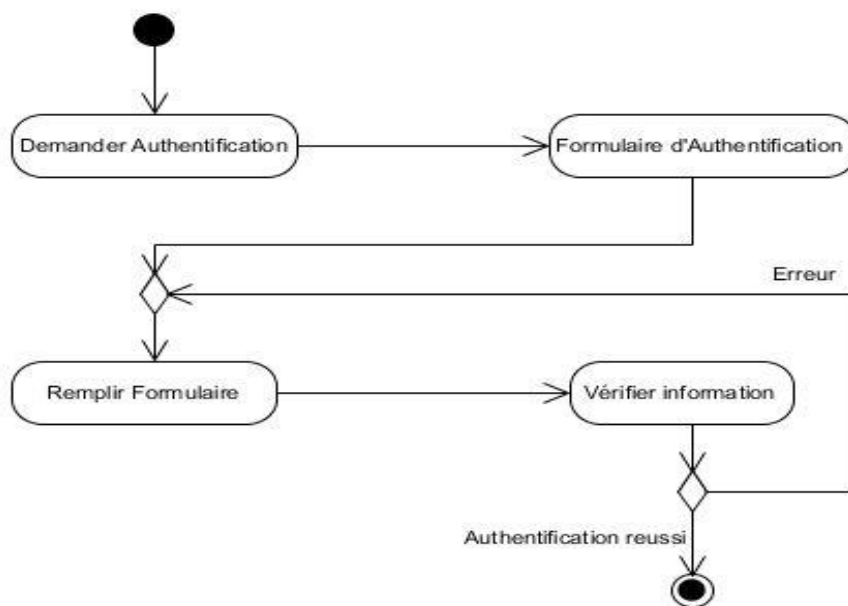


Figure 6.5 : Diagramme d'activité de l'authentification

Et pour notre application, son diagramme d'activité en globale est représenté dans la figure 6.6

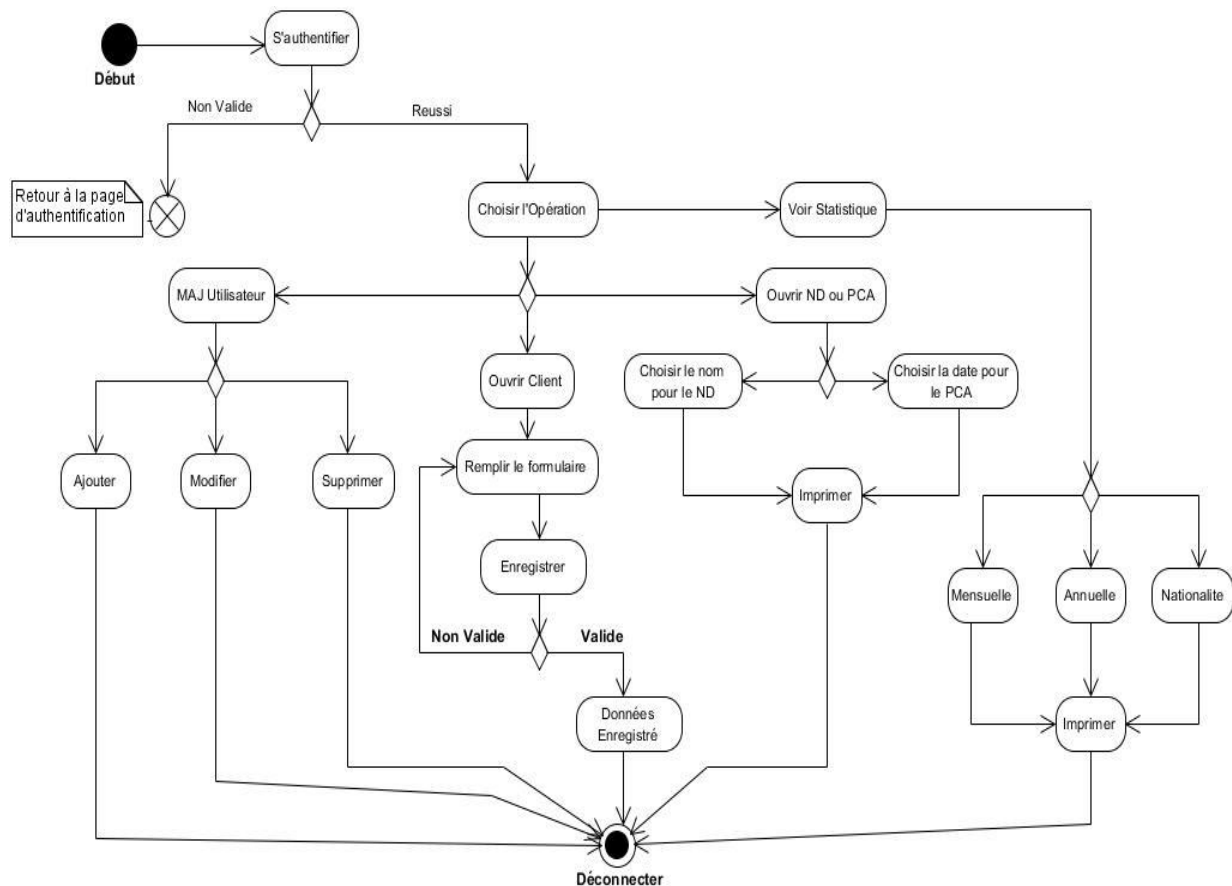


Figure 6.6 : Diagramme d'Activité Globale de l'application

6.6 Le diagramme de classe

Un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes [Hajalalaina A.R., 2016].

Les éléments formant un diagramme de classe sont recueillis dans le tableau, et peut être obtenu en combinant les précédents diagrammes partiels lors de la création des diagrammes de séquence. Une classe peut être décrite selon le tableau.

Tableau 6.3: Description d'une classe

Elément	Description
<p><u>Classe</u></p> <pre> classDiagram class NomClasse { -attribut : type +methode() : type_retourne } </pre>	<p>Objet avec des attributs et des opérations que l'on peut définir.</p> <p>Chaque attribut/opération se voit attribuer une encapsulation (+ pour public, - pour private) et un type ou un type retourné dans le cas des méthodes</p>

Le diagramme de classe de notre application est représenté dans la figure 6.7:

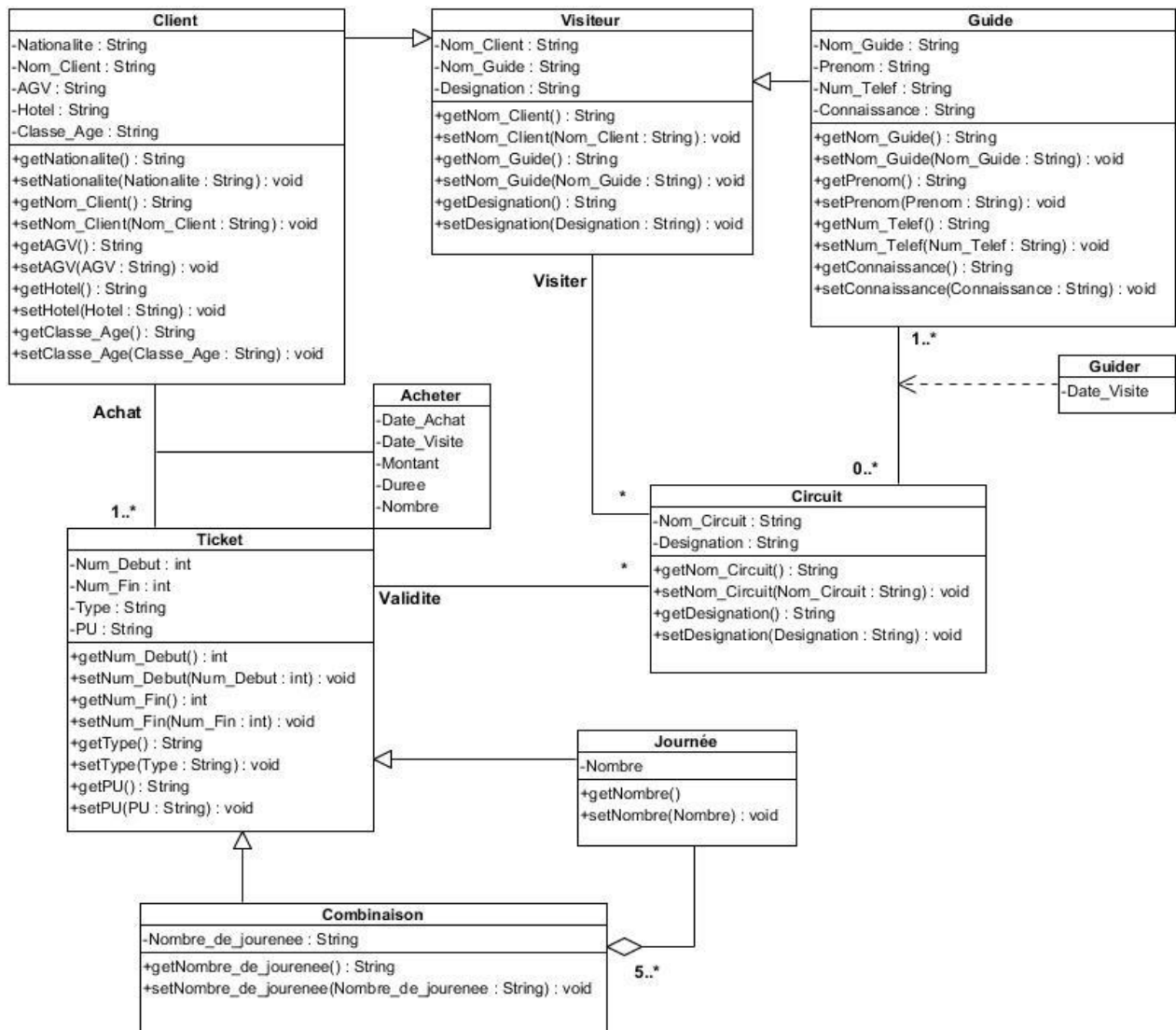


Figure 6.7 : Diagramme de classe de l'application

6.7. Conception des Interfaces Hommes-Machines

6.7.1. Phase de conception

Pour l'évaluation et support de communication entre les intervenants, d'où l'importance des outils de la maquette au prototype (à jeter ou réutilisable) vers le produit final.

6.7.1.1. Maquette

La maquette se divise comme suit :

- Ensemble des objets graphiques donnant une image de l'écran-utilisateur mais sans les fonctionnalités ;
- Support de communication entre les concepteurs (phase initiale) pas d'outil spécifique.

6.7.1.2. Prototype

Le prototype est utile pour :

- l'évaluation de fonctionnement ;
- Importance de la rapidité de développement ;
- Permet la spécification précise et définitive.

6.7.1.3. Le Produit finale

Le produit final est souvent dans un autre environnement de programmation.

La figure représente la phase de conception de l'IHM [Razafindragy., 2016].

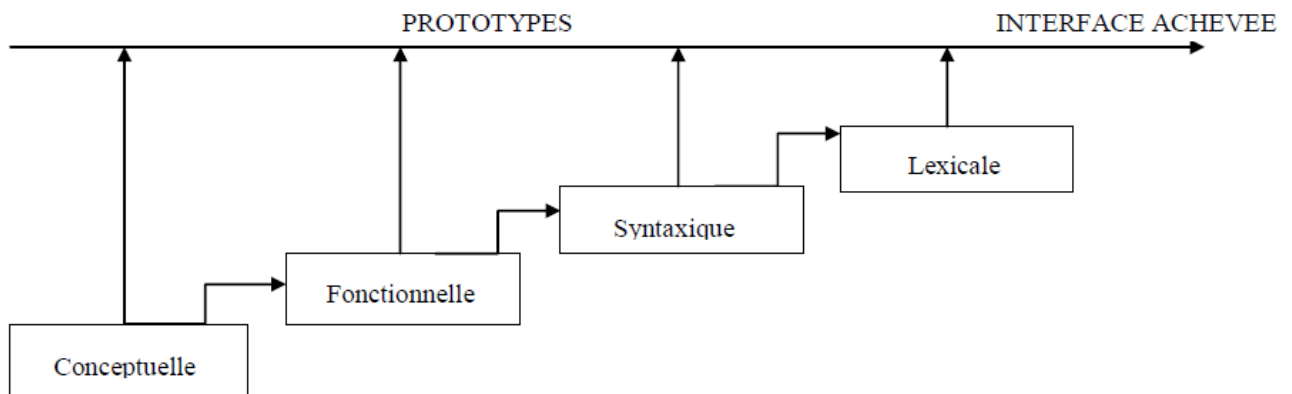


Figure 6.8: Phase de conception de l'IHM

6.7.2. Analyse préalable

Dans l'analyse préalable, il y a :

- Définir les objectifs du système : spécifier qualitativement et quantitativement les performances d'usage ;
- Analyse des tâches et activité utilisateur ;
- Identification des caractéristiques des utilisateurs.

6.7.3. Structure ou trame de la page

Nous devons représenter conceptuellement l'agencement des informations et le positionnement des différents objets : boutons, zone de contenus, image.

La structure de notre page est donc la suivante :

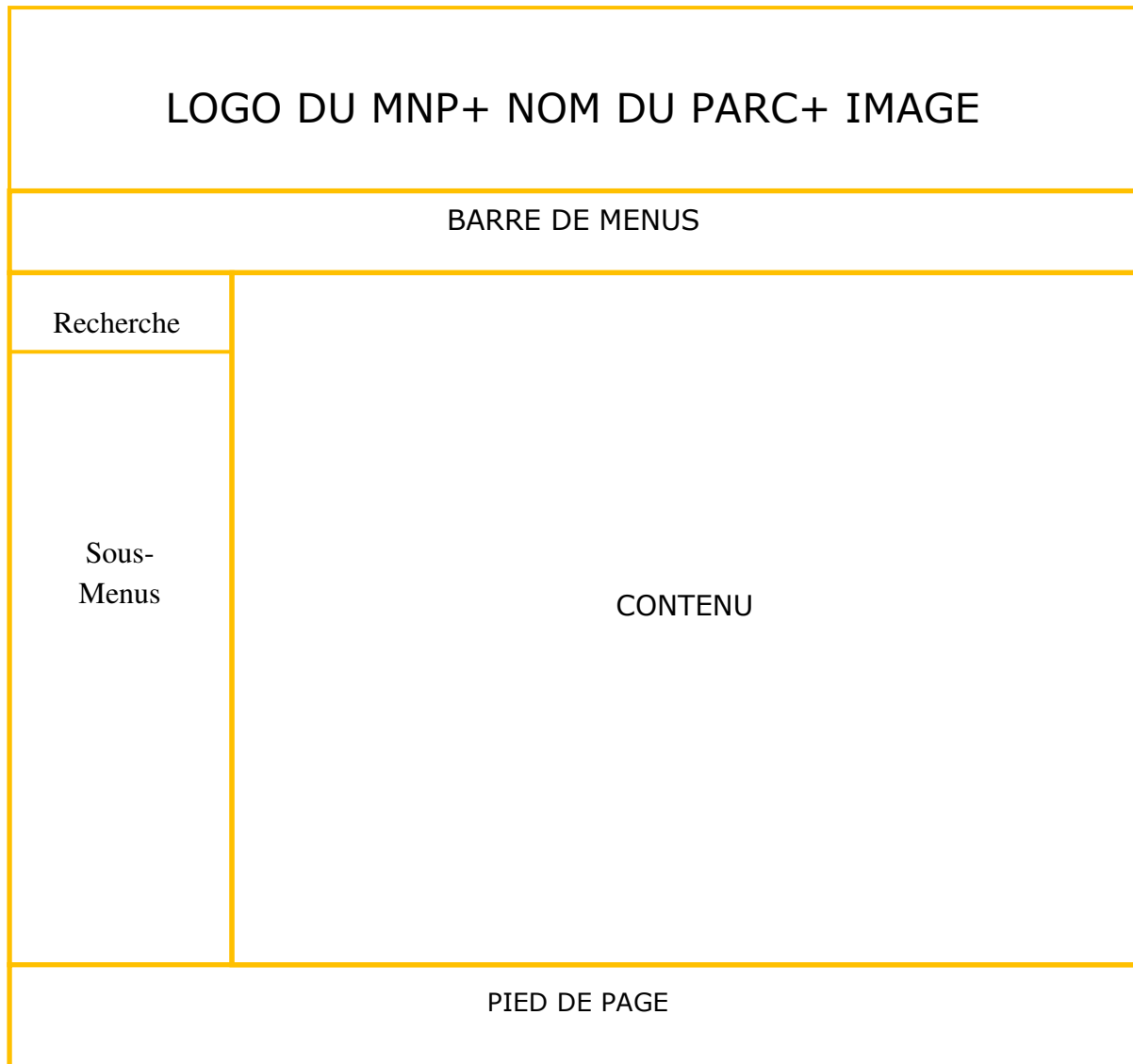


Figure 6.9 : Représentation de structure de page de l'Application

6.7.4 Aspect de l'application

La phase aspect de l'application [Crampes J.-B., 1997], appelé aussi Template graphique, consiste à définir les règles graphiques et ergonomiques permettant la sélection et la mise en page des informations à publier.

L'ergonomie d'une application web peut être définie par son capacité à répondre efficacement aux attentes des utilisateurs et à leur fournir un confort de navigation.

Les principaux critères d'ergonomie d'une application web sont les suivantes :

- Sobriété : simplicité, peu chargé ;
- Lisibilité : Clarté, organisation ;
- Accessibilité ;
- Adaptabilité : par exemple les manipulations des images par les internautes ;
- Rapidité : temps de chargement, image optimisées.

Après avoir achevé l'étape de conception de l'application, on va entamer dans ce chapitre et prendre la partie réalisation et implémentation qui permettent de s'assurer si le système est prêt pour être exploité par les utilisateurs en final.

A la fin de ce chapitre, les principes des technologies utilisés et les environnements de développements vont d'abord être présentés avant d'entailler la configuration de l'environnement et la mise en œuvre du développement. En fin, il y aura une brève présentation des interfaces graphiques de l'application web avec quelques imprimés écrans.

PARTIE III – REALISATION

Chapitre 7 : Présentation des outils de réalisation

7.1. Choix d'outils de développement

La nature du projet nécessite une conception d'une base de données pour modéliser la gestion. Le système de gestion a besoin d'une utilité de base donnée (SGBD), et un langage de programmation pour faire l'usage des données et une interface graphique facilitant l'emploi de l'application.

7.1.1. Choix du SGBD

7.1.1.1 Généralité

- Base de données :

Une base de données (BD) [Ullman L., 2004] est un ensemble d'informations, structurées, organisées (fichiers), partagé par plusieurs utilisateurs. Ces informations sont interrogées et mises à jour par l'intermédiaire d'un logiciel.

- Système de Gestion de la Base de données :

Le système de gestion de base de données [BD – Isabelle Comyn-Wattiau et Jacky Akoka ; 2010] est une suite de programmes qui manipule la structure de la base de données et dirige l'accès aux données qui y sont stockées.

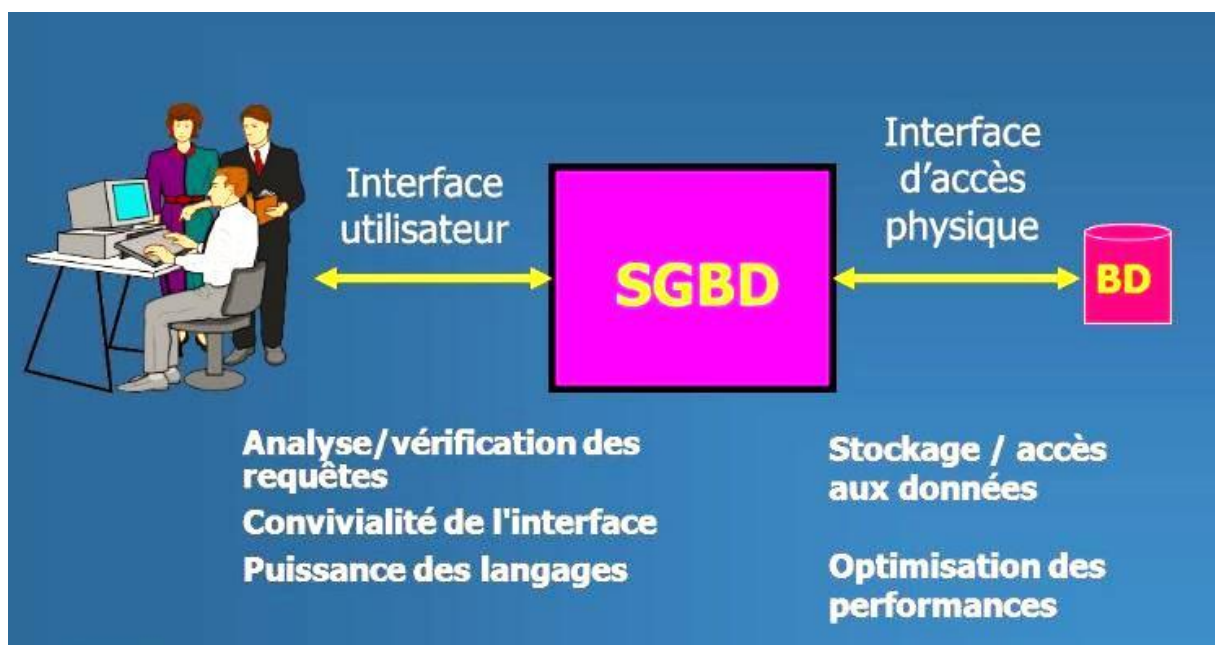


Figure 7.1 : Structure de la base de données [Jacky Akoka, 2010]

Une base de données est composée d'une collection de fichiers; on y accède par le SGBD qui reçoit des demandes de manipulation du contenu et effectue les opérations nécessaires sur les

fichiers. Il cache la complexité des opérations et offre une vue synthétique sur le contenu. Le SGBD permet à plusieurs usagers de manipuler simultanément le contenu, et peut offrir différentes vues sur un même ensemble de données.

7.1.1.2. Etude comparative des différents SGBD

Il existe plusieurs sortes du SGBD que nous pouvons utiliser dans le développement d'une application. Nous montrons dans le tableau 7.1 les caractéristiques des systèmes de gestion de base de données.

Tableau 7.1: Comparaison des SGBD [Rakotomalala T.M.D., 2016].

Base de données	Open source	Libre	Gratuit	Facile à manipuler	Simplicité
MS Access	Non	Non	Non	Oui	Oui
MySQL	Oui	Oui	Oui	Oui	Oui
PostgreSQL	Non	Oui	Oui	Non	Non
Oracle	Oui	Oui	Oui	Non	Non

Notre choix est orienté vers MySQL comme un SGBD car elle est un SGBD extrêmement attachant dans certaines conditions de rôle et de mise en production. Et elle permet de fournir un très grand nombre de requêtes et ce pour un coût très bas. Dans ce cas MySQL est infranchissable et peut fournir jusqu'à 2 fois plus de requêtes que tous les autres SGBD sur des machines semblables.

PostgreSQL présentent quelques difficultés au niveau de son utilisation, et d'un autre côté, Oracle présente une trop grande incompatibilité avec les capacités requises.

7.1.1.3. Présentation du SGBD MySQL

Nous avons utilisé le MySQL car il est un véritable serveur de base de données SQL Multiutilisateur et multithread. SQL est le plus populaire de base de données dans le monde. Le SQL (Structured Query Langage), un Langage d'Interrogation Structuré est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information.

a. Avantages

MySQL est un SGBDR édité par Oracle corporation. Il apporte divers avantages du fait que [Ullman L., 2004] [Jean-François P., Mars 2017 [en ligne]] :

- il est une solution très courante en hébergement public ;
- il effectue une très bonne intégration dans l'environnement Apache/PHP ;
- il est « open source » ;
- une version cluster existe depuis la version 4 ;
- un ordonnanceur et un partitionnement sont intégrés depuis la version 5.1 ;
- il est facile à déployer et à prendre en main ;
- il aménage de plusieurs moteurs de stockage configurables au niveau des tables et adaptés aux différentes problématiques.

b. Inconvénients

Le MySQL représente aussi quelques inconvénients [Ullman L., 2004]:

- il ne supporte qu'une faible partie des standards SQL-92 ;
- il ne maintient pas les triggers et les procédures stockées ;
- la gestion des transactions se fait, uniquement, avec les moteurs en Falcon ou InnoDB;
- il dispose de peu de richesse fonctionnelle ;
- il manque de robustesse pour de fortes volumétries de données ;
- il ne supporte pas l'héritage de tables ;
- il ne dispose pas de vue matérialisée.

c. Connexion à une base de données

Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données, pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données.

d. Le cheminement de données

Le cheminement de données se déploie comme suit :

- Un utilisateur se connecte sur une page.
- Il émet une requête afin d'obtenir l'affichage de la page.
- Le serveur obtient le message et le transmet au contrôleur suivant une route spécifique.
- La route elle, va par rapport à la page demandée, appeler un contrôleur et son opération.

- Le contrôleur traite alors l'action, et une fois terminée il répond à l'utilisateur avec la page requise.

7.1.2. Choix de langage de Programmation web

En informatique, un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent [Randrianirina F. P., 2014]. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, et de significations.

Au niveau du traitement et du développement, il y a des variétés des langages de programmation que nous pouvons utiliser car comme ledit le langage de programmation nous permet d'écrire notre application. Il joue le rôle d'une interface entre la base de données et l'utilisateur. Comme le SGBD, il y a beaucoup de langage de programmation.

Il existe plusieurs types de langage de programmation web comme le JSP, ASP, PHP. Nous allons voir quelques rôles de ces langages.

7.1.2.1 J.S.P

Le **Java Server Pages** ou **JSP** [Jean-Michel D., 2014[en ligne]] est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique.

La syntaxe du JSP ajoute des balises XML, appelées actions *JSP*, qui peuvent être utilisées pour appeler des fonctions. De plus, cette technique permet la création de bibliothèques de balises JSP qui agisse comme des extensions au HTML ou au XML. Les bibliothèques de balises offrent une méthode indépendante de la plate-forme pour étendre les fonctionnalités d'un serveur HTTP. Il existe aussi un langage de script particulier, appelé Expression Language (EL) destiné à réduire l'injection de code java au sein des pages JSP ainsi qu'à étendre les possibilités des taglibs, tel que la JSTL.

Les JSP sont compilées par un compilateur JSP pour devenir des servlets Java. Un compilateur JSP peut créer une servlet Java en code source Java qui peut à son tour être compilé par le compilateur Java, ou peut créer le pseudo-code Java interprétable directement. Dans les deux cas, il est bon de comprendre comment le compilateur JSP transforme la page

en servlet Java. Voir l'exemple de page JSP fourni en fin d'article, avec la servlet créée et la page HTML résultante.

7.1.2.2 ASP

ASP (Active Server Pages) [Introduction à ASP[en ligne]] est un standard mis au point par Microsoft en 1996 permettant de développer des applications Web interactives, c'est-à-dire dont le contenu est dynamique. Ainsi une page web ASP (fichier repérable par l'extension .asp) aura un contenu pouvant être différent selon certains paramètres (des informations stockées dans une base de données, les préférences de l'utilisateur) tandis qu'une page web « classique » (dont l'extension est .htm ou .html) affichera continuellement la même information.

ASP est en réalité une technologie, ou plus exactement un environnement de programmation, permettant de représenter sous forme d'objets les interactions entre le navigateur du client, le serveur web, ainsi que les connexions à des bases de données ou bien des composants COM (Component Object Model). Les ASP sont donc exécutées du côté du serveur et non du côté client (les scripts écrits en JavaScript ou les applets Java s'exécutent dans le navigateur de la personne connectée à un site).

Les ASP sont intégrables au sein d'une page Web en HTML à l'aide de balises spéciales permettant au serveur Web de savoir que le code compris à l'intérieur de ces balises doit être interprété afin de renvoyer des données au navigateur du client.

7.1.2.3 PHP

PHP [Jean-François P., Juin 2010 [en ligne]] est un langage interprété (un langage script) exécuté du côté serveur et non du côté client. Il permet de décrire et de créer des pages web à travers lesquelles l'utilisateur peut échanger des informations avec le serveur, c'est ce qu'on appelle des pages web dynamiques. Il permet aussi un affichage dynamique d'information, c'est-à-dire que le texte affiché peut dépendre de variables. La syntaxe du langage provient de celles du langage C, du Perl et du JAVA. Les instructions PHP sont généralement contenues dans des fichiers d'extension PHP. Ces fichiers peuvent contenir du HTML, entremêlé avec le code PHP. Quand un navigateur demande un tel fichier, le serveur Apache «exécute les instructions PHP, qui produisent une page HTML. Une fois la page html générée, le serveur la renvoie au navigateur, qui ne voit qu'une page HTML. Les principaux atouts au langage PHP sont les suivants :

- Une grandes communauté de développeurs partagent des certaines milliers d'exemple de scripts PHP ;
- La gratuité et la disponibilité du code script ;
- La disponibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML) ;
- La simplicité d'interfaçage avec des bases de données ;
- L'intégration au sein de nombreux serveurs web.

PHP permet un interfaçage simple avec de nombreux systèmes de gestion de bases de données (SGBD), parmi lesquels : Adabas D, dBase, Empress, FilePro, Informix, Interbase, mSQL, MySQL, Oracle, PostgreSQL, Solid, SyBase, Velocis, Unix, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes comme Unix, Linux, Windows et Solaris.

Pour pouvoir choisir un langage, on confronte le langage de programmation par leurs caractéristiques.

Voici un tableau comparatif entre les différents langages de programmation destiné au Web

Tableau 7.2: Comparaison des langages de développement web

LANGAGE WEB	PHP	JSP	ASP
Multiplateforme	OUI	OUI	NON
Gratuit	OUI	OUI	NON
Libre	OUI	OUI	NON
Simplicité	OUI	NON	NON

En se basant sur ce tableau et le choix de langage de programmation web précédant, nous avons tiré PHP comme étant le plus adapté au projet du fait qu'il dispose d'une facilité d'utilisation remarquable et aussi fiable pour le SGBD choisi.

Après avoir le choisi pour le développement, nous nous sommes demandé s'il serait mieux de construire l'application entièrement nous-même ou si un Framework pourrait simplifier une partie du développement. Il y a aujourd'hui un vaste choix de Framework PHP.

7.1.3. Comparaison des Framework PHP

7.1.3.1 Point Fort de framework PHP

Il convient avant tout de définir des critères avant d'élire le meilleur framework PHP [Nicolas F., 2013 [en ligne]]. En voici quelques-uns :

- **Prise en main** : dans tous les frameworks, il est nécessaire de passer quelques temps afin de le prendre en main. Cette période peut être plus ou moins longue (de quelques heures à plusieurs semaines) ;
- **Flexibilité** : le framework peut imposer des contraintes très fortes pour obliger tous les développeurs à développer de la même façon. Mais cela peut finir par être limitant ;
- **Taille des projets** : les petits projets peuvent être développés avec un framework ultra-light ;
- **Documentation** : une documentation complète est un réel plus ;
- **Performance** : certaines frameworks sont trop gourmands. Même le moindre « Hello World » peut nécessiter l'appel à plus de 100 fichiers différents ;
- **Communauté** : un forum actif sera synonyme de personne prêt à vous aider en cas de problème ;
- **Évolutivité** : du quand la date de la dernière mise à jour ?

Tous ces critères peuvent être plus ou moins pondérés en fonction de ce que vous attendez d'un framework. Pour ma part, je souhaitais un framework simple à prendre en main (en l'espace de quelques jours maximums), assez flexible et très bien documenté.

7.1.3.2 Les Framework PHP

Au moment de motoriser ses développements, toute entreprise cherche le moyen d'avoir la meilleure solution pour gérer son environnement [Nicolas F., 2013 [en ligne]]. Cela passe donc bien entendu par le choix d'un IDE, le choix d'un gestionnaire de version, mais surtout par le choix d'un Framework PHP. Celui-ci permet en effet de forcer les développeurs à utiliser des conventions dans leur façon de coder, et donc à n'importe quel développeur de pouvoir prendre en main l'ensemble du projet rapidement par la suite. Il permet aussi de guider les développeurs dans le choix technologique et d'architecture afin de pouvoir ajouter des « briques » au projet facilement par la suite.

Un Framework PHP répond normalement à de nombreuses demandes des développeurs. Il doit par exemple prendre en compte nativement les éléments suivants :

- **Modèle MVC** : modèle vue contrôleur. Il permet de structurer d'une application en distinguant la partie présentation, la partie base de données et la partie applicative.
- **Template** : gestion des gabarits. Il permet de séparer le code applicatif de la présentation.
- **Cache** : Il permet de stocker les pages afin d'optimiser leur temps de chargement.
- **Gestion des SGBDR** : Il doit pouvoir gérer plusieurs types de base de données. MySQL bien entendu, mais éventuellement PostgreSQL ou Oracle en fonction des besoins.
- **ORM** : Mapping de relation objet. Cela permet de gérer la base de données sous forme d'objets.
- **Scaffolding** : échafaudage. Permet de créer un espace d'administration d'un site sans aucun développement, uniquement à partir de l'ORM.
- **Conventions** : Oblige les développeurs à utiliser les mêmes conventions de codage afin d'avoir un code uniforme.
- **URL conviviales** : règles de redirection. Il doit pouvoir gérer les URLs facilement.

7.1.3.3 Les différents frameworks existants

Après étude, cinq (05) frameworks semblent régulièrement utilisés et le plus célèbre : Zend Framework, Symfony, CakePHP, CodeIgniter et Yii. En France, les deux plus utilisés sont très certainement Zend Framework et Symfony. Néanmoins les USA ou la Chine ont plutôt tendance à utiliser CodeIgniter ou Yii [Nicolas F., 2013 [en ligne]].

La courbe des recherches relative à ces frameworks durant les dernières années représente la figure suivante [Nicolas F., 2013 [en ligne]] :

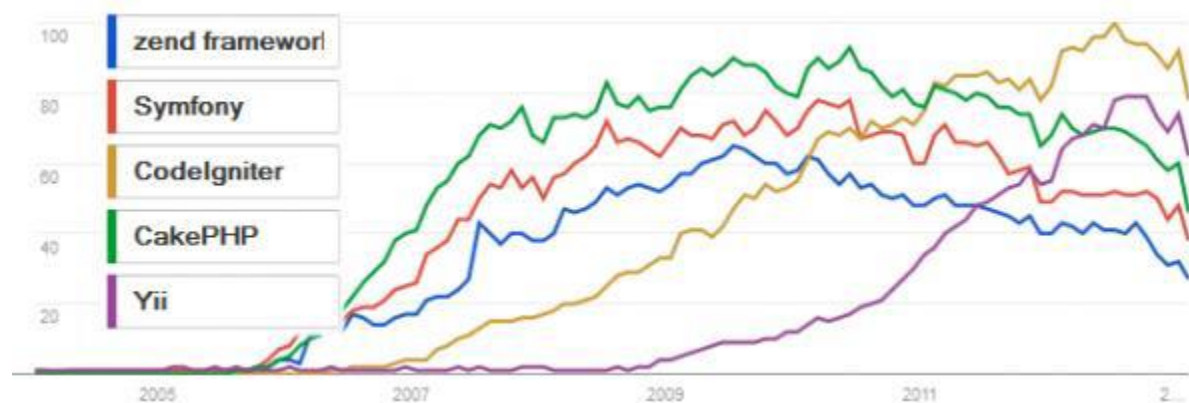


Figure 7.2: Courbe des recherches relatives à ce Framework PHP

On constate que ces frameworks ont décollé fin 2005, en même temps que Ruby on Rails. Le CakePHP a longtemps été le plus utilisé, mais CodeIgniter a pris les devants début 2011. Yii est actuellement en train de cartonner et pourrait même finir par dépasser CodeIgniter dans les prochaines années.

Les Framework apportent une architecture de base ainsi que des fonctionnalités déjà implémentées. Mais, tous utilisent une architecture Modèle Vue Contrôleur.

Donc, pour mieux comprendre ce dernier, la figure 7.3 nous dessine son architecture.

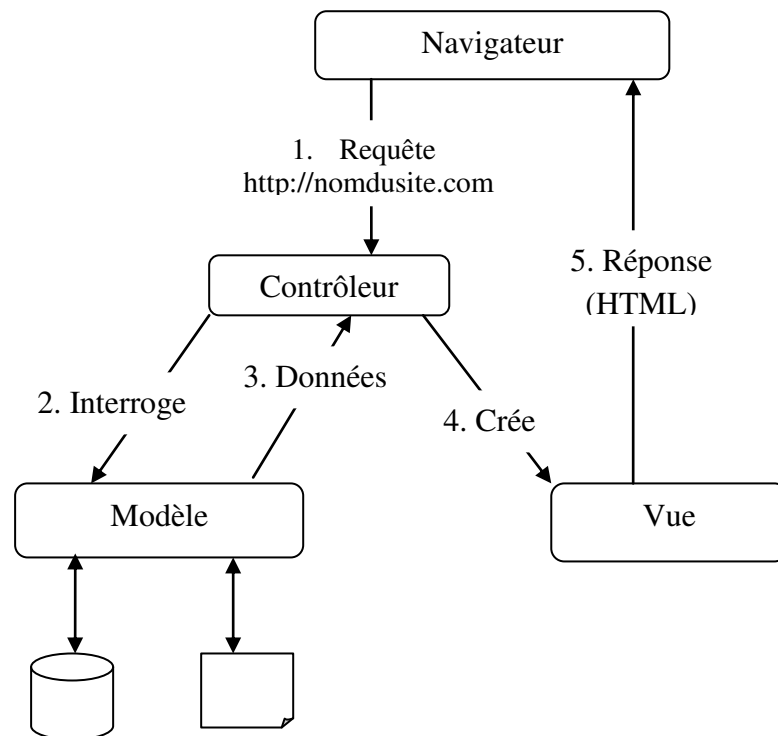


Figure 7.3 : Architecture MVC [adapté à partir de l'architecture MVC Wikipédia]

Dans le cas d'une application web, le contrôleur est appelé lorsque le navigateur envoie une requête au serveur pour connaître le contenu d'une page. Il interroge alors éventuellement la couche Modèle afin d'obtenir des données venant de diverses sources comme une base de données par exemple. Il crée ensuite une vue avec ces données, comme une page HTML, qu'il renvoie au navigateur.

La vue ne communique jamais directement avec la couche Modèle, et ni la vue ni le contrôleur n'accèdent directement aux sources de données. Chaque rôle (ou classe) est différencié, ce qui permet en cas d'évolution ou de maintenance, de ne toucher qu'à une partie du code sans altérer le reste du projet.

7.1.3.4 Choix de Framework

Le tableau 7.3 nous expose la différence entre quatre Framework le plus connues et le plus utilisés par les développeurs [Seralo, 2016 [en ligne]].

Tableau 7.3 : Comparaison des Framework PHP

Critère	CodeIgniter	Cake PHP	Zend Framework	Symfony
Dernière version	3.1.3	3.3.11	2.4.9	3.2.0
Facilité d'apprentis	Facile	Facile	Difficile	Difficile
Open source	Oui	Oui	Oui	Oui
Générateur CRUD	Oui	Non	Non	Oui
Simplicité	Oui	Non	Oui	Non
Sécurité	Oui	Oui	Oui	Oui
Economie espace	Oui	Non	Non	Non
Ouverture page	Très rapide	Un peu lent	Normale	Rapide

Après la comparaison et l'observation de ce tableau, CodeIgniter est celui qui nous a le plus séduit. En plus des éléments précédents, il ne nécessite en effet que très peu de configuration et permet de personnaliser tous les éléments du framework.

7.1.4 Présentation du Framework CodeIgniter

7.1.4.1 Définition

CodeIgniter est un framework MVC pour PHP4 et 5, inspiré de Ruby on Rails et distribué sous licence open source [Christophe C., 2011[en ligne]]. Il insiste sur son côté léger et très performant.

7.1.4.2 Installation

CodeIgniter se présente sous la forme d’une archive à extraire dans le répertoire du projet. La mise à jour n’est par contre pas aussi simple que dans les autres frameworks [Andrey A., 2011 [en ligne]]. La configuration de l’application n’étant pas dérivée d’une configuration globale, il est nécessaire d’appliquer manuellement certaines modifications lors de l’installation d’une nouvelle version.

7.1.4.3 Organisation des répertoires



- **/system/application** : l’application proprement dite : les contrôleurs, modèles, vues, bibliothèques utilisées et configuration de l’application se retrouvent dans leurs dossiers respectifs.
- **/system/codeigniter** : le Framework en lui même
- **/system/libraries** : les bibliothèques utilisées par CodeIgniter.
- **/system/plugins** : plugins installés
- **/system/scaffolding** : templates et images utilisées dans la génération des interfaces d’administration
- **/user_guide** : documentation, disponible en local.

7.1.4.4 Template

CodeIgniter utilise des templates au format PHP :

Template :

```
<table>
<tr>
<th>Id</th>
<th>Titre</th>
<th>Créé</th>
<th>Modifié</th>
</tr>
<?php foreach( $query->result() as $row ): ?>
<tr>
<td><?php echo $row->id?></td>
<td><?php echo $row->title?></td>
<td><?php echo $row->created; ?></td>
<td><?php echo $row->modified; ?></td>
</tr>
<?php endforeach; ?>
</table>
```

Figure 7.4 : Template CodeIgniter au format PHP

7.1.4.5 Accès aux données

CodeIgniter utilise une implémentation d'ActiveRecord.

Modèle :

```
<?php
class Items_model extends Model {

    function Items_model() {
        parent::Model();
    }
}
?>
```

Contrôleur :

```
<?php
class Comments extends Controller {

    function Comments() {
        parent::Controller();
    }

    function index() {
        $data['query'] = $this->db->get( 'comments' ) ;
        $this->load->view( 'comments_view', $data ) ;
    }
}
?>
```

Figure 7.5: Modèle et Contrôleur

Le scaffolding est activé par l'ajout d'un mot clé (configurable) dans l'url lors de l'accès à un modèle sans contrôleur. Un outil de débogage et de statistiques est intégré et permet d'obtenir les requêtes et le temps passé pour générer chaque page.

7.1.4.6 Cache

CodeIgniter dispose d'un système de cache assez basique. Une fois celui-ci activés, les pages sont mises en cache pendant une certaine période.

7.1.4.7 Documentation

La documentation de CodeIgniter est très complète [Andrey A., 2011 [en ligne]], mais comporte parfois des erreurs. Ainsi certaines parties du guide de démarrage rapide sont significativement différentes de celles contenues dans la documentation détaillée et ne permettront pas d'obtenir une application fonctionnelle.

7.1.4.8 Maturité

CodeIgniter est un framework plutôt jeune et dynamique, dont la première version date de février 2006. Assez complet, même si il manque des fonctions telles que la gestion des droits ou l'internationalisation. Conseillé pour les développements de petite envergure.

Voici un schéma qui présente le fonctionnement de CodeIgniter

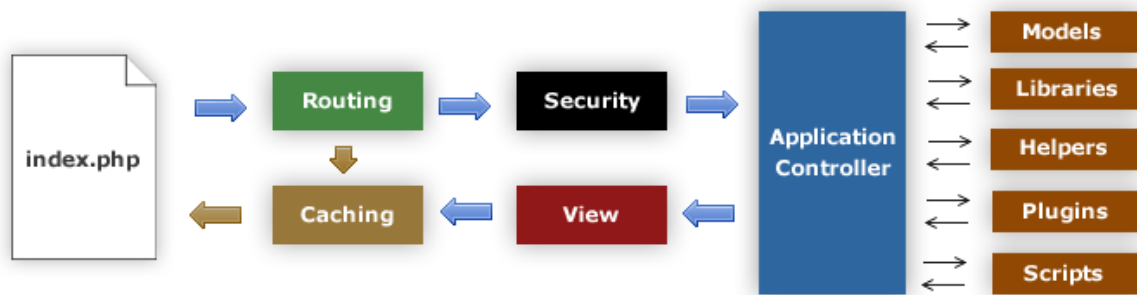


Figure 7.6 : Fonctionnement de CodeIgniter

On accède à toutes les pages de l'application via la page `index.php`. C'est elle qui nous redirige ensuite vers le contrôleur demandé.

Sur le schéma ci-dessus, c'est le routeur qui est chargé de décomposer l'URL en actions. Il est possible de modifier son comportement pour avoir des URL qui suivent une autre syntaxe.

Une fois le contrôleur trouvé, le système regarde si un fichier cache existe pour la page demandée, ce qui permet d'accélérer le chargement des pages lorsque leur contenu n'a pas changé.

Si la page n'est pas en cache, la requête HTTP est sécurisée, de même que les données transmises par l'utilisateur.

Ensuite seulement le contrôleur demandé est exécuté. Dans son code il est notamment possible de faire appel :

- A des modèles, pour accéder à des données.
- A des bibliothèques fournies par CodeIgniter, comme par exemple un système de sessions basé sur des cookies ou une bibliothèque d'uploader de fichiers. Il est aussi possible de créer ses propres bibliothèques.
- A des « helpers », qui sont des fonctions génériques pouvant être utilisées n'importe où dans l'application, par exemple pour faciliter la gestion des URL dans le code ou pour faciliter la gestion des dates. Là encore il est possible de créer ses propres helpers, et même de modifier ceux existants pour leur donner un comportement différent.

CodeIgniter est une structure de la source populaire, légère, ouverte écrite dans PHP et basé sur le MVC modèle architectural.

7.2 Présentation du langage et quelques outils de technologie complémentaire

7.2.1 Le langage HTML

Le HTML n'est pas un langage de programmation mais c'est un langage à balises, c'est un simple fichier texte contenant des balises permettant de mettre en forme le texte, les images, etc. Le HTML est un système qui formalise l'écriture d'un document avec des balises de formatage indiquant la façon où doivent être présentés le document et les liens qu'il établit avec d'autres documents.

7.2.2 Le langage CSS

Le CSS est appelé feuilles de style en cascade. Le concept de feuilles de style consiste à attribuer des caractéristiques de mise en forme à tout un groupe d'éléments de la page web. Nous les appelons feuilles de style en cascade car nous pouvons en définir plusieurs, et en cas de redondance de style, un ordre de priorité est donné par le browser (navigateur).

7.2.3 Le langage JAVASCRIPT

Le JavaScript est un langage de script incorporé dans un document HTML.

C'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. Ainsi le langage JavaScript est fortement dépendant du navigateur.

7.3 Outils de Technologie complémentaire (AJAX/JQuery)

Le *jQuery* est une source rapide, légère, ouverte la bibliothèque JavaScript conçue pour simplifier manipuler le HTML pagine et interactions d'Ajax. Quand a utilisé ensemble, ils forment une fondation puissante pour développer rapidement des applications et des applications web.

Le XML est utilisé dans AJAX pour la transmission des données entre le serveur et le client (c'est-à-dire le navigateur). Cependant, il est aussi possible que le serveur renvoie un résultat au format HTML, texte ou JSON. JSON (pour « **J**ava**S**cript **O**bject **N**otation », ou « notation objet JavaScript » en français) est un format de données utilisé en JavaScript pour décrire des objets.

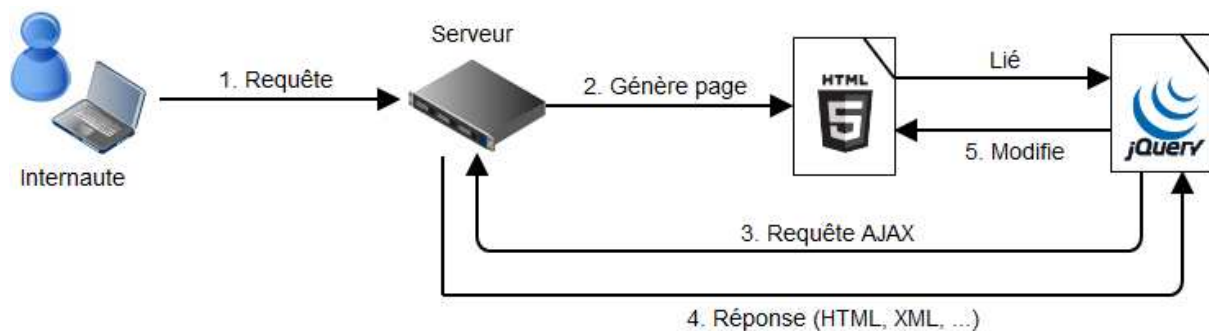


Figure 7.7 : Fonctionnement d'AJAX

Tout d'abord, l'utilisateur fait une requête pour obtenir le contenu d'une page de manière classique. Le serveur lui renvoie alors la page générée, qui est affichée dans le navigateur. Ensuite, en fonction de certaines actions (un clic, un déclenchement automatique, etc.), le code JavaScript réalise un appel AJAX vers le serveur via une URL. Cet appel est asynchrone, ce qui permet à l'utilisateur de pouvoir continuer à interagir avec la page affichée dans son navigateur. Lorsque le serveur renvoie les données demandées via la requête AJAX, le code JavaScript se charge de les traiter. On peut ainsi ajouter des informations sur une page sans avoir à la recharger. Tout est fait de manière invisible pour l'utilisateur.

7.4. Présentation des outils de développement

Cette phase consiste à représenter physiquement les données en utilisant un SGBD et un logiciel d'application pour gérer la base de données.

7.4.1. L'environnement du développement

L'environnement du développement comporte l'environnement matérielle et logicielle.

7.4.1.1. L'environnement matériel

Pour développer l'application, nous avons utilisé un ordinateur portable ayant comme caractéristique :

- Marque : Lenovo IdeaPad U530 Touch;
- Mémoire interne (RAM) : 8 Go ;
- Processeur : Intel(R) Core(TM) i7-4510U CPU @2.00GHz (4CPUs), ~2.6 GHz ;
- Disque dur : 320 Go ;
- Système d'exploitation : Windows 8.1 Professionnel 64 bits.

7.4.1.2. L'environnement logiciel

Lors de la conception et la réalisation de l'application, nous recourons sur :

- NetBeans IDE 7.3.1 ;
- WampServer 5.5.12.

a. *NetBeans IDE*

NetBeans [Qu'est-ce-que NetBeans[en ligne]] est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit (JDK) est requis pour les développements en Java.

b. *WampServer*

WampServer [Romain B., 2017 [en ligne]] est un serveur Apache, MySQL, PHP et Perl, X désignant le fait qu'il soit multiplateforme (cross-Platform en anglais), A pour Apache http Server, M pour MySQL, P pour PHP et enfin P pour Perl.

Chapitre 8 : Mise en œuvre du projet

8.1. La normalisation

Dans la normalisation, il vaut mieux d'abord définir le Web 2.0 et après nous allons découvrir l'ergonomie des interfaces hommes machines.

8.1.1. Web 2.0

Le Web 2.0 [O'Reilly T., 2005] a été introduit en 2004 par Tim O'Reilly pour décrire une nouvelle génération d'Internet centrée sur une participation plus active des internautes.

De façon générale, le Web 2.0 désigne une étape de l'évolution d'Internet. Le Web 1.0 propose du contenu consultatif et des communications par courrier électronique. Les gouvernements y contribuent en produisant de nombreuses pages sur la toile. Mais, fait important, l'internaute n'y joue qu'un rôle passif, car il ne peut pas apporter sa contribution à ce contenu [Hajalalaina A. R., 2016].

La figure 8.1 nous montre l'évolution du web [Dina N., 2014].

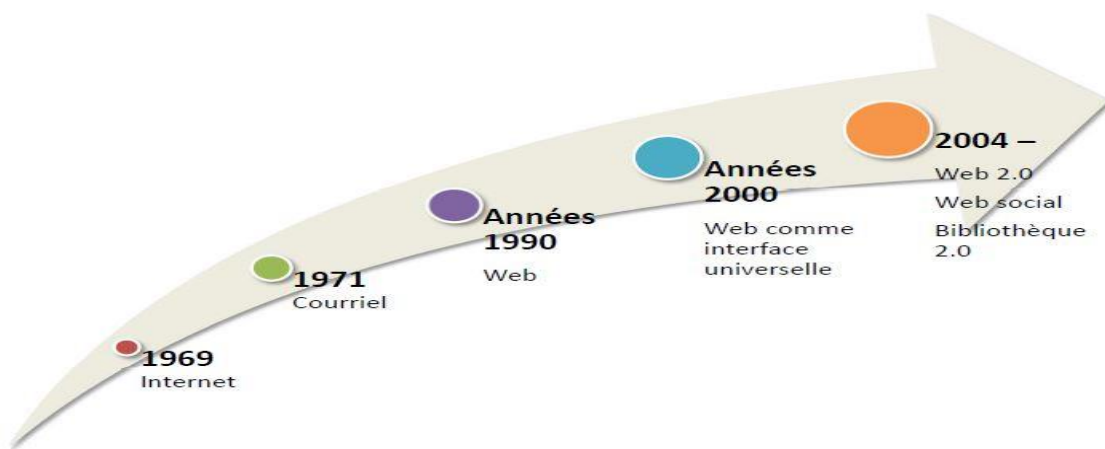


Figure 8.1 : Evolution du web

8.1.2. Interface Homme Machine

Cette définition se divise en deux éléments, à savoir, pour les ergonomes et les psychologues et pour les informaticiens concernés par la technologie de l'interaction.

- Pour les ergonomes et les psychologues :

L'Interaction Homme-Machine désigne l'ensemble des phénomènes physiques et cognitifs qui interviennent dans la réalisation de tâches informatisées.

- Pour les informaticiens concernés par la technologie de l'interaction :

Une interface est un assemblage de composants logiciels et matériels qui permet l'accomplissement de tâches avec le concours de l'ordinateur [Hajalalaina A. R., 2016].

8.1.2.1. Les perspectives de conception d'une IHM

La conception d'une interface utilisateur doit être effectuée suivant différentes perspectives, chacune d'elles ayant une influence sur la qualité de la conception globale.

- **Perspective fonctionnelle**

Elle vise à s'assurer que le système répondra aux objectifs pour lesquels il a été conçu et développé.

- **Perspective esthétique**

Elle vise à s'assurer que le système est plaisant dans son apparence et conforme à toute notion reconnue de la conception artistique (apparence visuelle de l'interface).

- **Perspective structurelle**

Elle vise à s'assurer que le système est conçu de sorte à en faciliter la maintenance et éventuellement l'extension [Dina N., 2014].

8.1.2.2. Les risques d'une mauvaise interface

Une mauvaise interface [Hajalalaina A. R., 2016] représente beaucoup de risques comme:

- Rejet des utilisateurs (technopathe) ;
- Coût d'apprentissage (formation) ;
- Utilisation incomplète (manque à gagner) ;
- Coût de maintenance ;
- Perte de productivité et de crédibilité !

8.1.3. Définition d'ergonomie

L'ergonomie (ou Human Factors) est la discipline scientifique qui vise la compréhension fondamentale des interactions entre les humains et les autres composantes d'un système, et la profession qui applique les principes théoriques, les données et les méthodes en vue d'optimiser le bien-être des personnes et la performance globale des systèmes. IEA (International Ergonomics Association) en 2000 [Sandrine BALBO., Septembre 1994] [J.-B. CRAMPES, 1997].

L'ergonome informatique s'intéresse à deux points :

- La structure interne de l'application.
- Son apparence, c'est à dire la façon de représenter la structure sous-jacente.

L'ergonome doit définir les informations qui interviennent dans chaque phase du dialogue et juger de leur utilité en fonction du contexte [Hajalalaina A.R., 2016].

8.2. Configuration minimale matérielle et logicielle

La configuration minimale des matérielles et logicielles est représenté sur le tableau suivante.

Tableau 8: Configuration minimale des matérielles et logicielles

Matérielles/ logicielles	Configurations
Matériel	Le développement d'une telle application requiert au minimum les matériels informatique suivant pour son bon déroulement mais aussi pour accélérer les taches dans le travail.
Utilisateur	<ul style="list-style-type: none"> – Disque dur, – RAM 512 Mo, – Ecran cathodique, – Clavier et souris.
Serveur	<ul style="list-style-type: none"> – Disque dur, – RAM 512 Mo, – Ecran cathodique, – Clavier et souris.
Logiciel	<ul style="list-style-type: none"> – Dreamweaver ; – NetBeans ; – WampServeur 5.1 ; – Mozilla Firefox 24.

8.3. Architecture Web

Cette section nous montre l'architecture logicielle et l'architecture matérielle de notre application en démontrant les rôles de chaque composant.

8.3.1. L'Architecture logicielle d'une application web

L'architecture d'une application web est la structure des structures (modules) d'un système.

« Elle inclut :

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

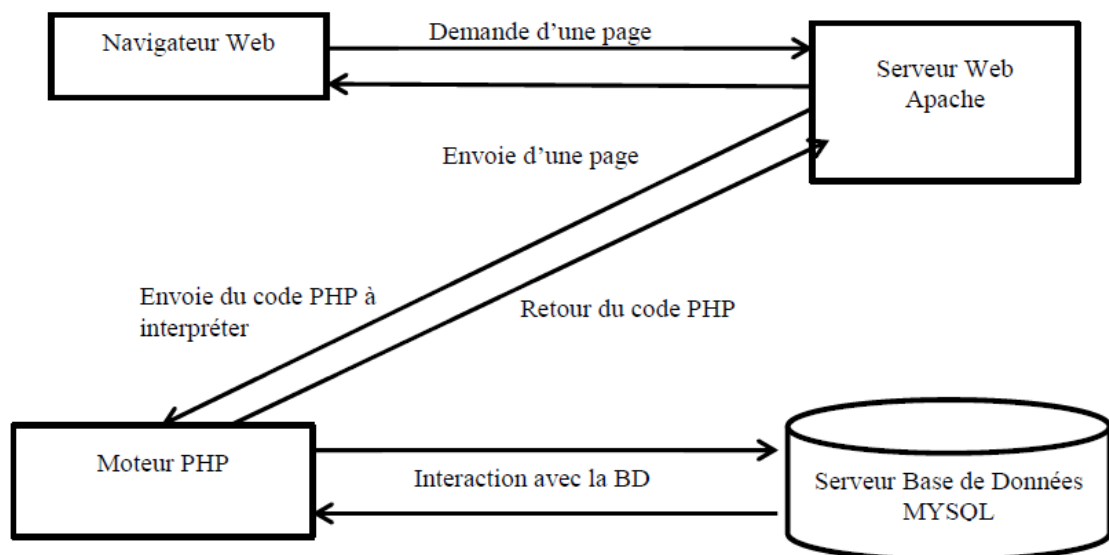


Figure 8.2 : L'architecture d'une application web

- Le **serveur web Apache** écoute les requêtes émises par la **Navigateur Web** (qui demandent des pages web) de chercher la page demandée et de la renvoyer.
- Le **moteur PHP** permet de décrire dans une page web, un affichage dynamique d'information. Quand la **Navigateur Web** demande un tel fichier, le serveur Apache exécute les instructions PHP, qui produisent une page HTML. Une fois la page HTML générée, le serveur la renvoie au navigateur, qui ne voit qu'une page HTML.
- Le **Serveur Base de Données MySQL** stock les données, sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL.

8.3.2. L'Architecture matérielle

L'architecture matérielle comprend toutes les caractéristiques générales, la conception, le choix et l'organisation des différents dispositifs électroniques des appareils informatiques.

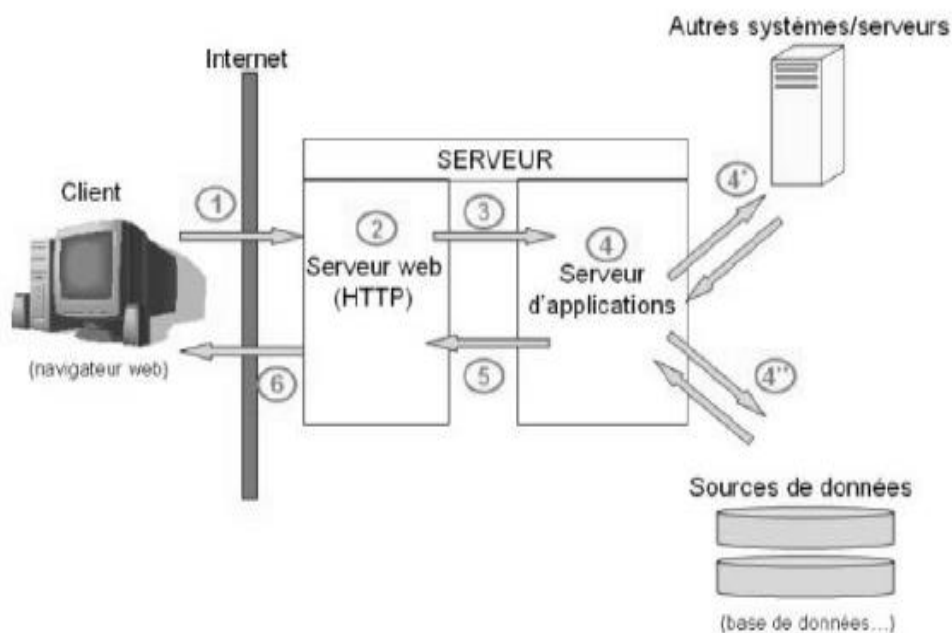


Figure 8.3 : Architecture matérielle

Pour utiliser correctement l'application en question, la figure nous montre la disposition de matériel requise pour que l'application soit utilisable [Rakotonirina V., 2014].

Ce schéma peut se résumer comme suit :

1. Le client émet une requête (Appelle une URL) pour demander une ressource au serveur. Exemple : `http://leserveur.com/home`. Il ne sait pas ici si la réponse qui va lui parvenir est statique (page HTML simple) ou dynamique (générée par une application WEB). Dans notre cas, il s'agit d'une application répondant à l'adresse home sur le serveur `leserveur.com`.
2. Côté serveur, c'est le serveur web (exemple : Apache) qui traite les requêtes HTTP entrantes. Il traite donc toutes les requêtes, qu'elles demandent une ressource statique ou dynamique. Seulement, un serveur HTTP ne sait répondre qu'aux requêtes visant des ressources statiques. Il ne peut que renvoyer des pages HTML, des images,... existantes.

3. Ainsi, si le serveur HTTP s'aperçoit que la requête reçue est destinée au serveur d'applications, il la lui transmet. Les deux serveurs sont reliés par un canal, nommé connecteur.
4. Le serveur d'applications (Apache) reçoit la requête à son tour. Il est, lui, en mesure de la traiter. Il exécute donc le morceau d'application (le contrôleur) auquel est destinée la requête, en fonction de l'URL. Cette opération est effectuée à partir de la configuration du serveur.
5. Une fois sa réponse générée, le serveur d'applications la renvoie, par le connecteur, au serveur web. Celui-ci la récupère comme s'il était lui-même allé chercher une ressource statique. Il a simplement délégué la récupération de la réponse, et celle-ci a été générée, mais ce n'est plus le problème.
6. La réponse est dorénavant du simple code HTML, compréhensible par un navigateur. Le serveur HTTP peut donc retourner la réponse au client.

Chapitre 9 : Présentation de l'application

Après avoir achevé la partie présentation des outils de développement et la mise en œuvre de l'application, il s'agit de présenter le résultat fini.

9.1. Fonctionnalités de l'application

Cette application représente beaucoup de fonctionnalités qui permettent d'améliorer la qualité du travail de gestion du service de vente à l'accueil actuel de Madagascar National Parks du PNI. Ce sont :

- Mise à jour des informations enregistrées ;
- Donner le PCA automatiquement par jour ou en demande sans avoir refaire le calcul ;
- Affichage des listes pour chaque enregistrement ;
- Imprimer la Note de débit ou bien la facture après avoir remplir le formulaire concernant ;
- Signal automatique au téléphone mobile du Chef de Volet Ecotourisme si quelqu'un ouvre l'application;
- Afficher la liste des guides ou des clients enregistrer;
- Ajout de nouveaux guides;
- Voir les statistiques ;
- Envoyer des informations par SMS mobile aux guides.

9.2. Présentation de quelques pages de l'application

Cette partie dénombre la présentation des Scénarios applicatifs de l'application. Nous allons présenter dans ce qui suit, les imprimés-écran des principales interfaces réalisées dans notre application web.

9.2.1. Page d'authentification

Si le page d'authentification s'affiche, le CV Ecot ou les deux Hôtesse tape son adresse mail et son mot de passe et puis il tape sur continuer. Le « Mot de passe oublié ? » est utilisé si l'utilisateur ne peut pas accéder à son compte ou il l'oublie mais seul l'administrateur qui peut l'avoir. En cliquant sur celui-ci, le page de vérification de leur email s'affiche, et si la vérification est validée, il peut changer son mot de passe.

La page d'authentification de notre application est représentée par la figure 9.1.



Veillez vous authentifier

Votre adresse e-mail

Votre mot de passe

Continuer

Copyright © 2017 by Parc National Isalo

Figure 9.1 : Page d'authentification

9.2.2. Page d'accueil

Pour la présentation, la page d'accueil sera affichée si l'utilisateur a réussi l'authentification. En première lieu, ça concerne la page d'accueil, c'est pour afficher la menu afin d'avoir l'utilisée. Après l'authentification, il a accès aux fonctionnalités de l'application. Ces derniers sont listés sur leur premier page.



Parc national Isalo
Commune Ranohira

ACCEUIL Agence de Voyage Facture PCA ou DEAP Guides Clients Statistiques Admin

Rechercher

Vous êtes ici : Accueil

BIENVENUE SUR CETTE APLLIATION WEB, SOYEZ ALÈSE

CLIENT

Malagasy
Etranger
Conventionné

CIRCUITS

Namaza
Piscine Naturel
Piscine Noire & Bleu
Cascade
Canyone de Maki &
Circuit 4 x 4

TICKET

Vente à Tana
Ticket Etranger
Ticket Malagasy

Le Parc National Isalo présente une beauté ancestrale unique due à ses vestiges archéologiques, ses sanctuaires culturels « Bara » (en particulier ses sites funéraires) et ses massifs ruiniformes sis dans une atmosphère magique faite de violents contrastes entre :

- L'aridité et le silence des falaises ocre où pousse une rare végétation rupicole (divers espèces de pachypodes, d'aloès, et d'euphorbes) écrasée par le soleil,
- Les appels de lémuriers (Prophithecus verreauxi, Lémur fulvus, Lémur Catta), des chants d'oiseaux (une cinquantaine d'espèces) et le doux murmure de l'eau qui résonnent dans les oasis de verdure le long des rivières et au fond des canyons.

Le PNI a vu naissance le 19 Juillet 1962 régit par le décret n°62.731, et c'était le WWF qui occupait sa gestion jusqu'à ce que l'Entreprise ANGAP a pris le relais. Depuis 1995, il a été sous contrôle de l'ANGAP qui a un dessein de protéger Isalo parce qu'il est une richesse exclusif à Madagascar et de conserver ses biodiversités, mais l'Entreprise a été nommé sous le titre de MNP à partir de l'année 2008 et jusqu'à aujourd'hui.

Localisation

Le parc Nationale Isalo est à :

683 Km au Sud-Ouest de Tananarive,
Environ 269 Km au Sud de Fianarantsoa

Il se trouve au province de Fianarantsoa, préfecture d'Ihosy et au Commune rurale de Ranohira.

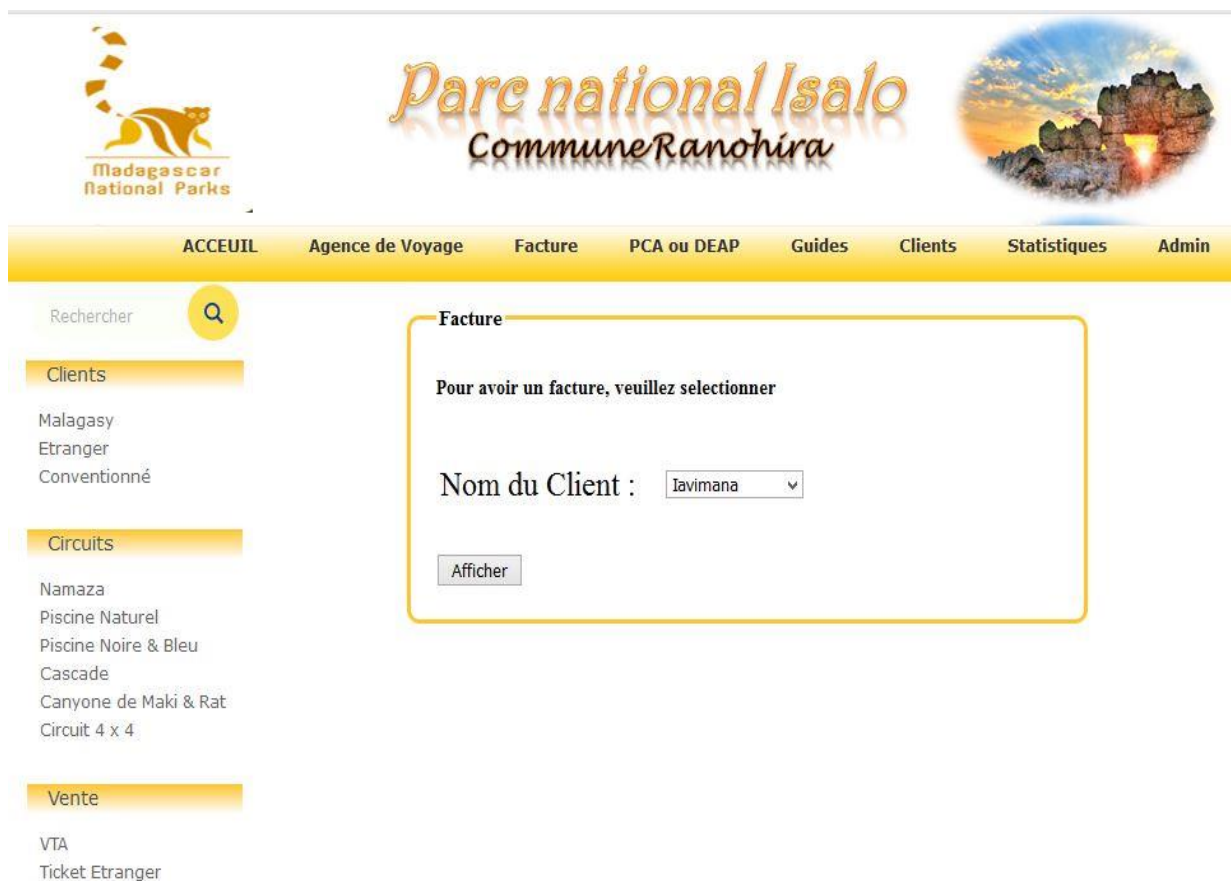
Il est limité au Sud par la RN7, son massif ruiniforme à l'Ouest, la rivière de Menamaty à l'Est et le massif lui-même au Nord.

Figure 9.2 : Page d'accueil de notre application

9.2.3 La Facture et la PCA

Sur la page d'accueil vous avez divers choix d'opérations, donc si vous avez choisi de cliquer sur la Facture, elle affiche une interface comme celui de la figure 9.3 en vous demandant de choisir le nom du client qui vous doit une note de débit ou facture.

De même pour le choix de la PCA, sauf que la seule différence est qu'avec le PCA, vous avez le choix de la date au lieu du nom.



The screenshot shows the website for Parc national Isalo, Commune Ranohira. The header includes the logo of Madagascar National Parks, the title 'Parc national Isalo Commune Ranohira', and a scenic image of the park. The navigation menu contains: ACCEUIL, Agence de Voyage, Facture, PCA ou DEAP, Guides, Clients, Statistiques, and Admin. The 'Facture' section is active, displaying a form with the title 'Facture' and the instruction 'Pour avoir un facture, veuillez selectionner'. The form contains a label 'Nom du Client :' followed by a dropdown menu currently showing 'Iavimana'. Below the dropdown is an 'Afficher' button. On the left side, there is a sidebar with a search bar and three main categories: 'Clients' (with sub-items: Malagasy, Etranger, Conventionné), 'Circuits' (with sub-items: Namaza, Piscine Naturel, Piscine Noire & Bleu, Cascade, Canyone de Maki & Rat, Circuit 4 x 4), and 'Vente' (with sub-items: VTA, Ticket Etranger).

Figure 9.3 : Interface pour le choix de nom du client à la facture

Si vous cliquer sur le bouton « Afficher » vous avez le résultat comme celle de la figure 9.4 suivante :

Madagascar National Parks
PARC NATIONAL ISALO

FACTURE

Nom du Client : lavimana

Date de visite	Type	Début	Fin	Nombre	PU	Montant
2017-04-04	AM	32897	32956	60	2000	120000
						Total: 120000

Arrêté à la somme de: Cent Vingt Mille Ariary

Payement en Espèce

Client

Hôtesse d'Accueil

Figure 9.4 : Forme de la facture à imprimer

Madagascar National Parks
PARC NATIONAL ISALO

PIECE COMPTABLE ACCEUIL

Date : 05/04/2017 RECETTE D.E.A.P

Note de Débit N°:

Type	Début	Fin	Nombre	PU	Montant
AE1	13501	13550	50	65000	3250000
AE2	661	700	40	110000	4400000
EM	13	20	8	500	4000
					Total:7654000

Arrêté à la somme de : Sept Millions Six Cents Cinquante Quatres Mille Ariary

CVECOT CVAF Secrétaire Hôtesse d'Accueil

Figure 9.5 : Forme à imprimer de la PCA

CONCLUSION

En guise de conclusion, nous pouvons dire que cette conception pratique a consommé notre formation théorique, dans cet apprentissage que nous avons appris beaucoup de nouvelles expériences et ça pourrait nous accompagner dans notre futur.

Ce projet de fin d'étude a traité de l'analyse du système de gestion à la conception. En effet on a entamé l'étude structurelle et fonctionnelle utilisant une conception orientée objet avec UML. Après l'analyse on a tiré le modèle de domaine et exprimé les besoins du MNP PNI dans des diagrammes de cas d'utilisation.

La création de cette application doit suivre des étapes bien précises avant d'être réalisée. L'expression des besoins se reporte à définir ce que l'on attend au système à mettre en place. Pour cela, le recueil de plus d'informations possibles auprès du demandeur est primordial, car ceci va nous permettre d'apprendre l'opportunité et la faisabilité du projet.

Le projet nous a permis en particulier d'être confrontés en pratique aux problématiques de modélisation qui est vraiment importante. Il nous a permis de s'adapter dans le monde professionnel. De plus, il nous a permis aussi d'approfondir les connaissances sur l'utilisation de la notation unifiée UML, et de maîtriser le Framework CodeIgniter.

Malgré cela, l'application que nous avons créée n'a pas atteint sa performance parfaite. Des améliorations doivent être envisagées afin d'augmenter cette exploit. Nous avons comme abords, de créer un site web pour que les clients puissent connecter en ligne pour faire la réservation et chercher de guide à leur besoins. De plus, une application avec quoi les clients peuvent acheter leurs tickets en utilisant leur carte de crédit.

Ce stage nous a été bénéfique soit au point de vue professionnelle que pédagogique. Elle nous a renforcé nos connaissances et aussi de découvrir de nouvelle horizon dans la vie quotidienne.

BIBLIOGRAPHIE

- [Andrey A., 2011], Installation CodeIgniter [en ligne]. British Columbia Institute of Technology. Disponible sur : < [http:// www.codeigniter.com/user_guide/installation/index.html](http://www.codeigniter.com/user_guide/installation/index.html) > (consulté le : 15 Février 2017) ;
- [Bakari M., 2016], *Support de cours conduite de projet informatique*, Troisième année en Développement d'Application et Système d'Information, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
- [Bakari M., 2016], *Support de cours de PHP*, Troisième année en Développement d'Application et Système d'Information, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
- [Christophe C., 2011], Apprendre CodeIgniter de A à Z [en ligne]. TUTORA. Disponible sur : < http://www.tutora.fr/apprendre-codeigniter_ > (consulté le : Janvier 2017) ;
- [Crampes J.-B., 1997]. *Interfaces graphiques ergonomiques – Conception et Modélisation*, Ellipses, collection TECHNOSUP / Génie Logiciel ;
- Document *fiche de poste de MNP*, 20 Janvier 2017 ;
- [Dina N., 2014] [Ullman L., 2004,], *conception et réalisation d'une application pour la gestion d'assurance automobile au sein de la cmar ny havana, agence centrale de fianarantsoa*, Mém. Licence. Département Informatique, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
- Etude comparative des différents cycles de vie [en ligne]. Disponible sur : < <http://www.over-blog.com/article-etude-comparative-des-différents-cycles-de-vie-de-logicielle> > (consulté le : Fevrier 2017) ;
- [Hajalalaina A. R., 2016] [Anne-Marie H., 2013], © 19/12/02 2-13, *Génie logiciel*, , *cours Génie Logiciel*, Troisième année en Développement d'Application et Système d'Information, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
- [Hajalalaina A. R., 2016] [Jean-François P., 2014,], *cours IHM*, Troisième année en Développement d'Application et Système d'Information, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
- Introduction à ASP[en ligne]. The Best Cours. Disponible sur : < <http://www.bestcours.com/318-introduction-asp.net-2010> > (consulté le : Fevrier 2017) ;

-
- [Jean-François P., 2014,] *Ergonomie d'une application web* [en ligne]. Comment Ça Marche. Disponible sur : < <http://www.commentcamarche.net/contents/1255-ergonomie-d-une-application-web> > (consulté le : Janvier 2017) ;
 - [Jean-François P., 2010,] *PHP – Introduction* [en ligne]. Comment Ça Marche. Disponible sur : < [http:// www.commentcamarche.net/contents/1351-php-introduction](http://www.commentcamarche.net/contents/1351-php-introduction) > (consulté le : Février 2017) ;
 - [Jean-François P., 2017], *Base de données et systèmes de gestion de base de données (ou SGBD)* [en ligne]. Comment Ça Marche. Disponible sur : < <http://www.commentcamarche.net/contents/104-bases-de-donnees-introduction> > (consulté le : Janvier 2017) ;
 - [Jean-Michel D., 2014,] *Développons en Java- Les JSP (Java Server Pages)* [en ligne]. Développons en Java. Disponible sur : < <http://www.jmdoudoux.fr/java/dej/chap-jsp.htm> > (consulté le : Février 2017) ;
 - [Lai M., 1998.] *Penser objet avec UML*, InterÉditions ;
 - [Laurent A., 2010], « UML 2 » : pour le concept d'UML ;
 - [McDermid J. et Ripken K., du 1984.] *Life cycle support in the ADA environment*. University Press;
 - [Mohamed A. D., 2011,] *Conception, développement et intégration d'une application embarquée de téléchargement des applications android ftab store*, Mém. maîtrise. Université virtuelle de Tunis ;
 - [Muller P.-A., 1998.] *Modélisation objet avec UML*, Eyrolles ;
 - [Nicolas F., 2013,] *Comparatif des Framework PHP* [en ligne]. In Media Veritas, La vérité est dans le web. Disponible sur : < <http://www.inmediaveritas.com/comparatif-framework-php/> > (consulté le : 23 Février 2017) ;
 - [O'Reilly T., 2005,] « *Comment le web 2.0 améliorera les services aux citoyens ?* » ;
 - [Pierre G., 2005.] *Structure d'un Cahier des Charges*, Université de Paris 13 - IUT Villetaneuse ;
 - [Rakotonirina V., 2014], *Informatisation de la gestion de banque de sang*, Mém. Licence. Département Informatique, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
 - [Randrianirina F. P., 2014], *Conception et réalisation d'un site web dynamique de l'école normale supérieur*, Mém. Licence. Département Informatique, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
-

-
- [Razafindragy A.J.A., 2016], *Refonte du site web de la Région Anôsy*, Mém. Licence. Département Informatique, Centre Universitaire de Formation Professionnalisante / Université de Fianarantsoa ;
 - [Romain B., 2016.] Démarrez avec wampserver [en ligne]. WampServer. Disponible sur < <http://www.wampserver.com/#begin-wrapper> >. Consulter le 02 Mars 2016.
 - [Rumbaugh J., Jacobson I. et Booch G., 1999]. *The Unified Modeling Language Reference Manual*, Addison-Wesley;
 - [Seralo, 2016.] PHP Framework comparaison [en ligne]. Social Compare. Disponible sur : < <http://socialcompare.com/fr/comparison/php-frameworks-comparison> > (consulté le : Février 2017) ;
 - [Sylvie V., 2010.] *Cycle de vie du logiciel et bonnes pratiques de développement* ;
 - [Sarra R.A, 2013] *Conception et développement d'une application mobile de vente flash sous androïde*, Ecole supérieur des sciences économiques et commerciales Tunis. Licence appliquée en informatique de gestion 2013 ;
 - [Ullman L., 2004,] « MySQL Développement de site web » .

ANNEXE

ANNEXE

INSTALLATION DU FRAMEWORK CODEIGNITER

1. Installation



Nous allons voir ensemble comment mettre en place le framework.

Pré requis :

- Il est nécessaire d'avoir un serveur Web en place sur votre ordinateur. Voici plusieurs tutoriels pour en mettre un en place selon votre système d'exploitation (Linux / Windows)
- Penser à activer le module Apache suivant :
 - **Rewrite_module** (Linux / Windows)
- Ensuite, créons un Virtual Host pour notre projet. (Linux / Windows)
- Pour finir, installer votre IDE (Notepad++, Eclipse, ...)

A l'intérieur de l'archive, on trouve le répertoire CodeIgniter. Ce répertoire comprend :

- Le répertoire **application**, qui sera le répertoire de notre application
- Le répertoire **système**, qui est le cœur du Framework
- Le répertoire **user_guide**, qui est le guide offline de CodeIgniter

On décompresser ces 3 répertoire que l'on vient mettre dans notre répertoire d'application.

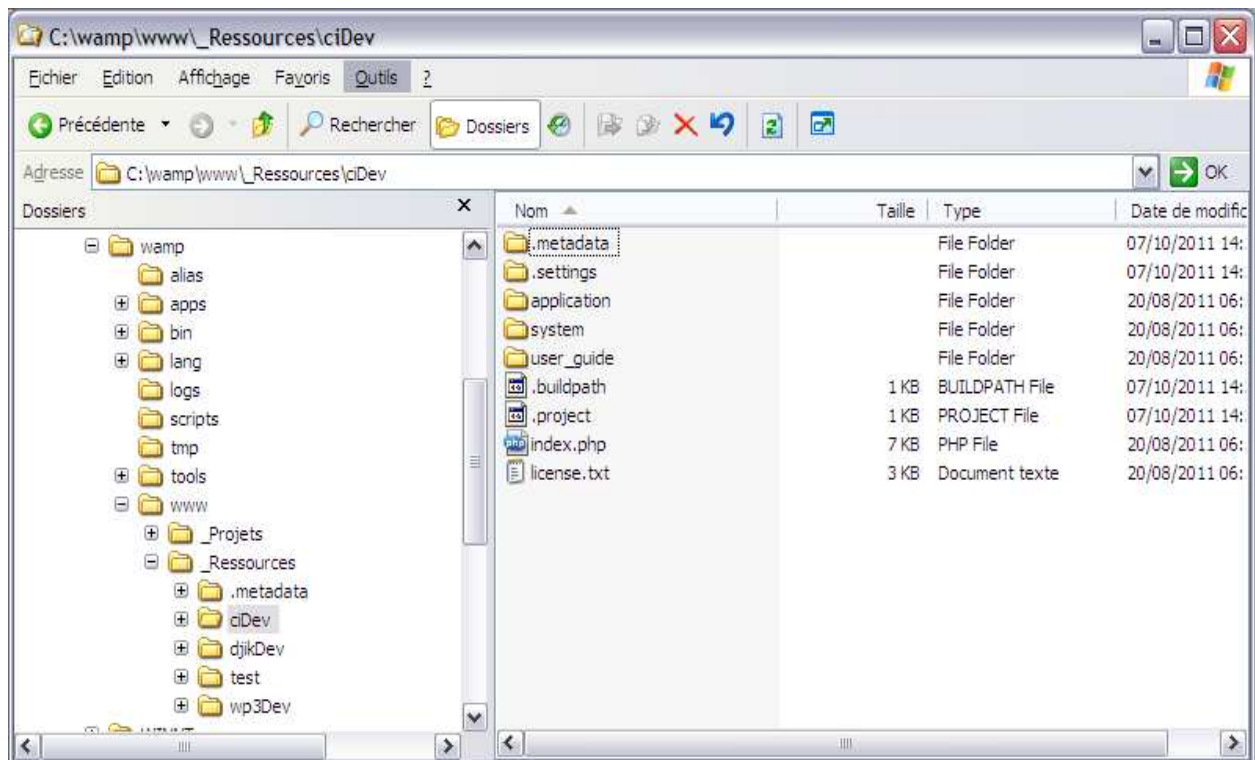


Figure A.1 : Répertoire d'application

- On lance notre navigateur

L'index.php s'exécute en premier lieu.

Maintenant, vous allez ouvrir le fichier index.php. La première chose à configurer est de définir si le framework est actuellement en période de développement, de test ou de production. La conséquence immédiate est l'affichage ou non des erreurs. Normalement, la constante est déjà sur développement.

```
<?php                                     define('ENVIRONMENT',
'development')//(development | testing | production) ?>
```

Une fois cela fait, vous pouvez modifier les deux variables suivantes : **system_path** et **application_folder**. Le premier est le chemin vers le dossier system et le second est celui vers le dossier application. La configuration par défaut est fonctionnelle. On pourra par la suite les changer sans problème.

```
<?php
/*SYSTEM FOLDER NAME */
$system_path = "system";
/*APPLICATION FOLDER NAME */
$application_folder = "application";
?>
```

La documentation de CodeIgniter recommande de renommer ces dossiers pour éviter à un visiteur mal intentionné de connaître leur emplacement. On pense que vous pouvez vous en passer dans un premier temps.

Vous avez fini d'installer CodeIgniter. Vous avez normalement accès à la page d'accueil du framework (Comme la mienne, c'est <http://localhost/Isalo/>) et celui-ci vous souhaite la bienvenue.

Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

```
system/application/views/welcome_message.php
```

The corresponding controller for this page is found at:

```
system/application/controllers/welcome.php
```

If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).

Page rendered in 0.0085 seconds

Figure A.2 : Page d'accueil application CodeIgniter

Si vous ne voyez pas ce message, reprenez bien les étapes dans l'ordre et assurez-vous de bien avoir les prérequis de votre version téléchargée (version minimale de PHP, droits de lecture...).

2. Mise en place de l'arborescence

Maintenant que le framework est installé, attardons-nous sur l'arborescence de CodeIgniter.

– Les dossiers **application** et **system**

À côté de votre `index.php`, vous avez deux dossiers : **application** et **system**. Le dossier `system` contient *tous les codes natifs de CodeIgniter* alors que le dossier `application` va contenir *tous vos codes*. Vous ne devrez donc jamais modifier les fichiers présents dans le dossier `system`, et on va vous en expliquer la raison tout de suite.

Imaginons que vous n'aimez pas le nom d'une méthode. Vous allez donc lancer une petite recherche et modifier le nom de cette méthode. Sauf que vous allez avoir de gros problèmes.

En voici trois qui méritent d'être cités.

- Si une nouvelle version de CodeIgniter sort, vous allez devoir refaire vos manipulations. Et ne dites pas que vous ne ferez pas la mise à jour, car vous allez risquer des problèmes de sécurité.
- En modifiant les sources, vous êtes susceptibles de laisser passer des failles, car il y a de fortes chances que vous ne connaissiez que l'aspect « utilisateur » du framework.
- Un des avantages de l'utilisation d'un framework est la facilité de travailler à plusieurs, car chacun sait comment le système fonctionne. Mais si vous modifiez les sources de CodeIgniter, son comportement ne sera plus le même.

Nous verrons par la suite comment effectuer ces changements sans toucher à ce dossier.

- Le dossier **assets**

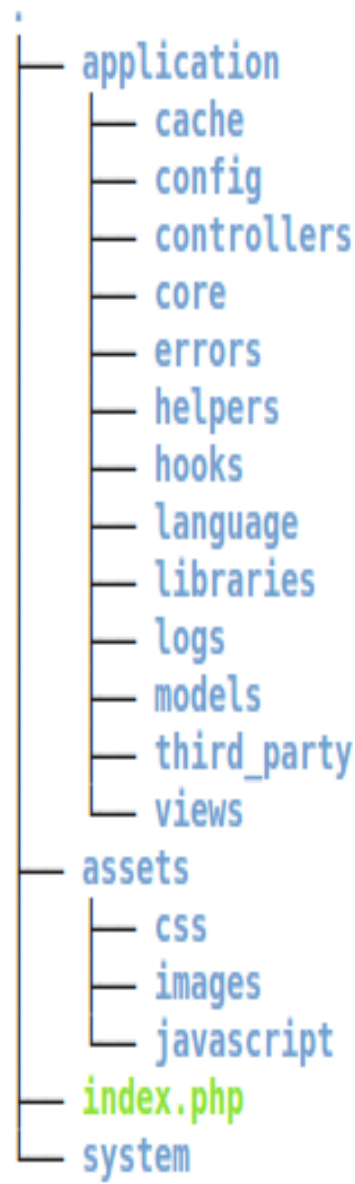
Avant de décrire précisément le contenu du dossier application, vous allez créer un dossier assets, toujours dans le même dossier que le fichier index.php. Et à l'intérieur de celui-ci, créez aussi les dossiers css, javascript et images.

Vous disposez maintenant d'une arborescence très claire.

- **assets** contiendra toutes les ressources qui vont être téléchargées par les visiteurs. Par exemple, les feuilles de style externes (CSS), les fichiers JavaScript et les images. Cependant, vous pouvez aussi ajouter des dossiers tels que musiques, videos, xml (pour les flux RSS, les animations flash). C'est comme si vous alliez faire un copier-coller de ce dossier dans l'ordinateur de vos visiteurs. Rien ne doit être confidentiel.

- **system** contient les sources de CodeIgniter, vous ne devez pas y toucher.
- **application** contiendra tous vos scripts, qu'il faudra placer dans les bons dossiers. C'est ce que nous allons voir tout de suite.
- **Le dossier application**

Voici l'arborescence du dossier application avec les explications des dossiers dont nous nous servirons dans la prochaine partie.



- **config** : ce sont des fichiers permettant de configurer CodeIgniter ou une bibliothèque. Certains sont inclus automatiquement, d'autres seulement lorsque vous les demandez.

- **controllers** : ce dossier contiendra tous nos contrôleurs.

- **errors** : ce sont les pages d'erreurs. Libre à vous de les personnaliser selon votre design.

- **helpers** : c'est le répertoire pour vos helpers. Les helpers fournis avec CodeIgniter sont situés dans le dossier system.

- **hooks** : c'est un dossier qui contient des fichiers assez complexes. Ils permettent d'exécuter des scripts à différents moments du processus d'exécution de CodeIgniter.

- **language** : ce répertoire contiendra tous vos fichiers de langue dans le cas où vous souhaitez un site internationalisé.

- **libraries** : nous placerons nos bibliothèques dans ce dossier. Comme les helpers, les bibliothèques fournies par CodeIgniter sont situées dans un autre dossier.

- **models** : le répertoire des modèles.

- **views** : le répertoire des vues.

En respectant les emplacements des différents fichiers, vous gagnerez en clarté.

RESUME

Nous avons accompli notre stage au sein du Madagascar National Parks Parc National Isalo au service d'accueil dans lequel nous avons conçu et réalisé un projet de développement d'une application web intitulé « Administration de vente au service d'accueil du parc national Isalo ».

Durant ce stage, malgré les contraintes de temps, on a réussi à concevoir l'analyse et une partie de l'application et surtout on a eu des expériences professionnelles.

Dans ce document se trouve tous les détails concernées, depuis l'étude de l'existant jusqu'à la réalisation de l'application, comme la présentation des outils utilisés et les modélisations du système. Nous avons comparés les outils utilisés par rapport à d'autres et étudiés l'architecture de l'application après avoir modélisé les besoins de l'utilisateur. L'application s'est aboutie avec une interface facile à manipuler et les fonctionnalités requis par la production que nous avons illustrés dans des captures d'écran.

Mots-clés : PHP, PostgreSQL, Marchandise, Multitâche, Note de débit, SGBD, Tickets.

ABSTRACT

We accomplished our practicum within Madagascar National Parks Isalo National Park to the service of welcome in which we conceived and achieved a project of development of an application web which have a title" Administration of trade to the service of welcome of the Isalo National Park ".

During that workshop, in spite of the constraints of time, we succeeded in conceiving the analysis and a part of the application and especially we got professional experiences.

In this document are all concerned details, since the survey of the existing until the realization of the application, as the presentation of the tools used and the modeling of the system. We compared the tools used in relation to other and studied the architecture of the application after having specify the user's needs. The application resulted to itself with an easy interface to manipulating and functionalities required by the operators that we illustrated in captures of screen.

Keywords: PHP, PostgreSQL, Merchandise, Multitask, Note of debit, SGBD and Tickets.