



UNIVERSITE DE FIANARANTSOA

ECOLE DE MANAGEMENT ET D'INNOVATION TECHNOLOGIQUE

Mention : INFORMATIQUE

Parcours : Développement d'Application Internet et Intranet (DAII)

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

**CONCEPTION ET REALISATION D'UNE APPLICATION DE
TRAITEMENT DES INCIDENTS ET DEMANDES LIEE A
L'EXPLOITATION DU TELEDECLARATION DE L'IMPOT
SYNTHETIQUE (IS) A FIANARANTSOA**

Présentée par : NAMBININTSOA Volatantely

Sous l'encadrement de TSIORINANTENAINA René

Année universitaire 2023-2024

CIRRICULUM VITAE

ETAT CIVIL

NOM : NAMBININTSOA

PRENOM : Volatantely

Né le 29 Janvier 2002 à Mananara-Nord

ADRESSE : Près IVK 21 Ankazobe Ambany Fianarantsoa

Tel : 034 71 082 99

E-mail : volatantelynambinintsoa@gmail.com



FORMATIONS

Ecole de Management et d'Innovation Technologique (EMIT)

- Présent : Troisième année de licence en Développement d'Application Internet et Intranet
- 2022-2023 : Deuxième année de Licence en Développement d'Application Internet et Intranet
- 2021-2022 : Première année de Licence en Développement d'Application Internet et Intranet

Facultés de Médecine Fianarantsoa

- 2019-2020 : Première année en Médecine Humaine

Lycée Publique Mananara-Nord

- 2019-2020 : Terminale D , Baccalauréat série D

COMPETENCES EN INFORMATIQUE

Bureautique : Microsoft : Word , Excel , Access et PowerPoint

SGBD : MySql, PostgreSQL

Méthode de conception : MERISE

Langage de modélisation : UML

Langage de programmation : Python, PHP, Java, JavaScript

Librairies et Framework : ReactJs, NodeJs, SpringBoot, Laravel

Technologie Web : HTML, CSS

Système d'exploitation : Windows, Linux

Réseaux : Configuration Switch et Routeur, Câblage

EXPERIENCE PROFESSIONNELLE

Stagiaire Informatique (2 mois)

DREN HAUTE MATSIATRA

- Conception et réalisation de système de gestion des résultats des examens CEPE au sein de la DREN HAUTE MATSIATRA

CONNAISSANCES LINGUISTIQUES

LANGUE	COMPREHENSION	Expression Orale	Expression Ecrite
FRANÇAIS	Bien	Bien	Bien
ANGLAIS	Moyenne	Moyenne	Moyenne
MALAGASY	Très Bien	Très Bien	Très Bien

AVANT PROPOS

Dans le cadre de nos études dans la mention Développement d'Application Internet et Intranet (DAII) à l'école de Management et d'Innovation Technologique (EMIT), les étudiants en troisième année de licence doivent effectuer un stage de trois mois au sein d'une entreprise en vue d'obtention du diplôme de Licence Professionnelle. Ce stage aura pour objectif la mise en pratique des connaissances théoriques acquises en classe, mais également l'adaptation au milieu du travail et confirmer l'adéquation des connaissances aux demandes du marché réel.

Lors du stage, il est également demandé de rédiger un ouvrage portant sur le stage effectué en guise de rapport de fin d'étude, d'où ce rapport de stage.

Nous avons passé notre stage dans la Direction Régionale des Impôt à Fianarantsoa (DRI) et notre thème se pose sur « LA CONCEPTION ET REALISATION D'UNE APPLICATION DE TRAITEMENT DES INCIDENTS ET DEMANDES LIEE A L'EXPLOITATION DU TELEDECLARATION DE L'IMPOT SYNTHETIQUE (IS) ». Ce stage m'a ainsi permis de vivre le quotidien dans la vie professionnelle et aussi un avantage qui peut être très utile dans la formation de cadre supérieur qui a été confié.

REMERCIEMENTS

Ce travail a pu être effectué grâce à la franche collaboration et la bonne volonté de plusieurs personnes auxquelles nous adressons notre profonde reconnaissance.

Nous tenons à remercier :

-Dieu tout Puissant, pour l'amour qu'Il nous porte chaque instant de notre vie, pour tous ce qu'Il nous donne, et pour la réalisation de tous ceci.

- Monsieur Le Professeur HAJALALAINA Aimé Richard, Président de l'Université de Fianarantsoa pour sa très haute bienveillance à notre égard.

-Monsieur Le Professeur RAKOTONIRAINY Hasina Lalaina, Directeur de l'Ecole de Management et d'Innovation Technologique (EMIT) de m'avoir donné l'opportunité d'acquérir les connaissances nécessaires et d'avoir consenti au fait de nous envoyer en stage cette année.

-Madame RABEZANAHARY Hobiniaina, Chef de mention Informatique à l'Ecole de Management et d'Innovation Technologique (EMIT) d'avoir collaboré avec nous pour la réalisation de ce projet.

-Madame RANDRIANARISOA Voahangiarivelo, Directeur de la Direction Régionale des Impôts d'avoir bien voulu valider ma candidature pour ce stage à l'institut.

-Monsieur RANDRIANANTENAINA Henri Arthur, mon encadreur professionnel, pour sa patience et ses judicieux conseils pour la réalisation de ce stage.

-Tous les enseignants et personnels de l'Ecole de Management et d'Innovation Technologique (EMIT)

-Tous les personnels de la Direction Régionale des Impôts (DRI)

-Mes Parents pour leurs soutiens moral et financier, leurs encouragement et inestimable aide durant la réalisation de ce projet.

-Enfin, nos amis qui nous ont toujours soutenu et encouragé dans la réalisation de ce projet et de ce présent ouvrage.

LISTE DES FIGURES

Figure 1.1: Organigramme de l'emit	7
Figure 2.1: organigramme du dri\hm	10
Figure 2.2: organigramme du cf/hm.....	13
Figure 4.1: modele en cascade	24
Figure 4.2: modele en v	25
Figure 4.3: modele en spirale	26
Figure 4.4: modele par increment	27
Figure 4.5: diagramme uml	31
Figure 4.6: figure de l'acteur	32
Figure 4.7: FIGURE REPRESENTANT LE CAS D'UTILISATION.....	32
Figure 4.8: diagramme de classe	34
Figure 4.9: diagramme de classe en relation	35
Figure 6.1: diagramme de cas d'utilisation de gestion des traitements	44
Figure 6.2: le diagramme de sequence globale de l'administrateur.....	46
Figure 6.3: le diagramme de sequence global de l'utilisateur	47
Figure 6.4: diagramme de classe du projet.....	48
Figure 7.1: navigateur et javascript	52
Figure 7.2: popularite des technologies frontends	54
Figure 7.3: architecture d'un projet reactjs	55
Figure 7.4: structure mvc	58
Figure 8.1: architecture logicielle.....	65
Figure 8.2: architecture materielle	66
Figure 8.3: maquette ihm de l'application	67
Figure 9.1: page d'authentification	68
Figure 9.2: page du tableau de bord	69
Figure 9.3: liste des utilisateurs	70
Figure 9.4: liste des incidents	70
Figure 9.5: diagramme de l'analyse.....	71
Figure 9.6: liste des supports.....	71
Figure 9.7: liste des incidents du cote utilisateurs.....	72
Figure 9.8: liste des support du coté utilisateur.....	72
Figure 9.9: page de discussion	73

LISTE DES TABLEAUX

Tableau 1.1: TABLEAU RECAPITULATIF	4
Tableau 3.1: planning du stage.....	18
Tableau 4.1: comparaison des modeles de cycle de vie	27
Tableau 4.2: tableau representant les elements du diagramme de cas d'utilisation.....	32
Tableau 4.3: description de chaque ligne du diagramme de sequence.....	33
Tableau 4.4: diagramme de classe.....	34
Tableau 4.5: differents types de relations.....	35
Tableau 4.6: avantages et inconvenients de l'uml	37
Tableau 5.1: existant materiel de l'entreprise	38
Tableau 6.1: representation de dictionnaire de donnees.....	41
Tableau 6.2: diagramme de sequence d'importation de fichier	45
Tableau 6.3: diagramme de sequence d'authentification.....	46
Tableau 7.1: comparaison de reactjs et angular	53
Tableau 7.2: comparaison de symfony et laravel	57
Tableau 7.3: comparaison de mysql et postgresql.....	61
Tableau 7.4: comparaison de visual paradigme et staruml	63

LISTE DES ABREVIATIONS

AES	: Administration Economique et Sociale
CSS	: Cascading Style Sheets
CUFP	: Centre Universitaire de Formation Professionnalisant
DAII	: Développement d'Application Internet et Intranet
DGI	: Direction Générale des Impôts
DRI	: Direction Régionale des Impôts
EMIT	: Ecole de Management et d'Innovation Technologique
HTML	: HyperText Markup Language
IHM	: Interface Homme Machine
IS	: Impôt Synthétique
MVC	: Model View Controller
RPM	: Relation Publique et Multimédia
SGDB	: Système de Gestion de Base de Données
SQL	: Structured Query Language
UML	: Unified Modeling Language

GLOSSAIRE

Application : c'est un ensemble de programmes permettant de fournir des fonctions spécifiques (Par exemple : traitement de texte, traitement de vidéo...)

Base de données : c'est un ensemble de données organisées de telle manière qu'on puisse les exporter.

Langage de programmation : le langage de programmation est une conventionnelle destinée à formuler des algorithmes et produire des programmes.

Logiciel : c'est un ensemble d'éléments informatiques qui permettent d'assurer une tâche ou une fonction

MySQL: My Structured Query Language. C'est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture. Il est multi- thread et multiutilisateurs.

SGBD : Un Système de Gestion de Base de Données est un ensemble de logiciels qui sert à la manipulation des bases de données. Il sert à effectuer des opérations ordinaires telles que consulter, modifier, construire, organiser, transformer, copier, sauvegarder ou restaurer des bases de données.

Système d'Information : c'est l'ensemble organisé de ressources matériels personnels, données et une procédure permettant d'acquérir, traiter, stocker, communiquer les informations (Sous forme de données, texte, image, son)

SQL : Langage de programmation dédié à la manipulation de base de données à base de requête.

Système d'information : C'est l'ensemble organisé de ressources matériels personnels, données et une procédure d'acquérir, traiter, stocker, communiquer les informations (sous formes de données, texte, image, son).

Requête : Commande répondre à une syntaxe précise permettant la manipulation d'information à l'intérieur d'une base de données.

Télé-déclaration : Processus permettant aux contribuables de déclarer leurs obligations fiscales en ligne, sans déplacement physique, via une plateforme numérique dédiée.

Impôt Synthétique (IS) : Type d'impôt forfaitaire simplifié destiné aux petites entreprises et travailleurs indépendants, basé sur une estimation des revenus.

Gestion des incidents : Processus de suivi, de diagnostic et de résolution des problèmes techniques ou fonctionnels dans un système ou une application.

Gestion des demandes : Processus de traitement et de réponse aux requêtes d'assistance ou de modification provenant des utilisateurs d'une application ou d'un service

TABLE DES MATIERES

CIRRICULUM VITAE.....	I
AVANT PROPOS.....	III
REMERCIEMENTS	IV
LISTE DES FIGURES	V
LISTE DES TABLEAUX	VI
LISTE DES ABREVIATIONS	VII
GLOSSAIRE.....	VIII
TABLE DES MATIERES	X
INTRODUCTION GENERALE	1
PARTIE I : PRESENTATION GENERALE	2
Chapitre 1 : Présentation de l'Ecole de Management et d'Innovation Technologique (EMIT).....	3
1.1 : Historique.....	3
1.2 : Missions	4
1.3 : Formations existantes.....	4
1.4 : Cycle Licence	4
1.4.1 : Mention « Management », Parcours « Administration Economique et Sociale »....	5
1.4.2 : Mention « Informatique », Parcours « Développement d'Application Internet Intranet » et Parcours « Conception, Intégration et Gestion des Systèmes d'Information »	5
1.4.3 : Mention « Relations Publiques et Multimédia », Parcours « Communication Multimédia » et « Relations Publiques et Communication Organisationnelle ».....	5
1.5 : Cycle Master	6

1.5.1 : Mention « Management », Parcours « Management Décisionnel ».....	6
1.5.2 : Mention « Informatique », Parcours « Système d'Information, Géomatique et Décision » et Parcours « Modélisation et Ingénierie Informatique »	6
1.6 : Organigramme de l'EMIT	7
Chapitre 2 : Présentation de la Direction Régionale des Impôts (DRI)	8
2.1 : DRI/HM	8
2.1.1 : Compétences territoriales et techniques de la DRI/HM	8
2.1.2 : Historique du Service Régional des Entreprises Haute Matsiatra.....	8
2.1.3 : Compétence du Service Régionales de l'Entreprises	8
2.1.4 : Mission de la Direction Régionale des Impôts Haute Matsiatra	9
2.1.5 : Organigramme de la Direction Régionale des Impôts Haute Matsiatra.....	10
2.2 : Centre Fiscale Haute Matsiatra	13
2.2.1 : Organigramme de CF	13
2.2.2 : Compétence territoriale	13
2.2.3 : Technique Fiscal.....	14
2.2.4 : Organisation du personnel au sein de la CF/HM.....	14
Chapitre 3 : Présentation du projet.....	16
3.1 : Contexte	16
3.2 : Présentation de Traitement des incidents et demandes liée à l'exploitation du télé- déclaration IS :	16
3.2.1 : Définition.....	16
3.2.2 : Les différentes types de problèmes ou des incidents peuvent survenir.....	16
3.3 : Objectifs	17
3.4 : Origine du projet :	17
3.5 : Problématique du projet :	17
3.6 : Planning du stage :	18
PARTIE II : ANALYSE ET CONCEPTION DU PROJET	20
Chapitre 4 : METHODE ET NOTION UTILISEE.....	21
4.1 : Méthode utilisée.....	21
4.1.1 : Généralité sur le génie logiciel	21
4.1.1.1 : Modèle de cycle de vie.....	21

4.1.1.2 : Les étapes d'un cycle de vie.....	21
4.1.2 : Modèles de cycle de vie de développement logiciel	22
4.1.2.1 : Modèle en cascade	23
4.1.2.2 : Modèle en V	24
4.1.2.3 : Modèle de cycle de vie par incrément.....	26
4.1.3 : Choix du modèle de cycle de vie.....	27
4.2 : Le langage Unified Modeling Language	29
4.2.1 : Histoire et origine de l'UML	29
4.2.2 : Définition.....	30
4.2.3 : Diagrammes UML.....	30
4.2.3.1 : . Diagramme de cas d'utilisation.....	31
4.2.3.2 : Diagramme de séquence	33
4.2.3.3 : Diagramme de classes	34
4.2.4 : Avantages et inconvénients d'UML	37
Chapitre 5 : ANALYSE DU PROJET.....	38
5.1 : Etude préalable	38
5.2 : Analyse de besoins.....	38
5.3 : Analyse de l'existants :	38
5.4 : Etude de faisabilité :	39
5.5 : Evaluation de faisabilité :	39
5.5.1 : Faisabilité organisationnelle :	39
5.5.2 : Faisabilité technique :	39
5.5.3 : Faisabilité économique :	39
5.6 : Les critiques de l'existant :	39
5.6.1 : Niveau informationnel :	39
5.6.2 : Niveau organisationnel :	39
5.6.3 : Niveau techniques :	40
5.7 : Les solutions proposées :	40
Chapitre 6 : CONCEPTION DU PROJET.....	41
6.1 : DICTIONNAIRE DES DONNEES.....	41
6.1.1 : Types de dictionnaires du donnés :	42
6.2 : Les règles de gestions :	42

6.3 : Le diagramme de cas d'utilisation du système :	43
6.4 : Les Diagrammes de séquences :	45
6.4.1 : Le Diagramme de séquence d'importation de fichier :	45
6.4.2 : Le Diagramme de séquence d'authentification :	46
6.5 : Le Diagramme de classes :	48
 PARTIE III : REALISATION DU PROJET	 49
 Chapitre 7 : SPECIFICATION DES OUTILS DE REALISATION	 50
7.1 : Présentation des technologies utilisées	50
7.2 : Technologie frontend	50
7.2.1 : HTML	50
7.2.2 : CSS	51
7.2.3 : Javascript	51
7.2.4 : Choix de l'outil	52
7.2.5 : Présentation de ReactJS	54
7.3 : Technologie backend	55
7.3.1 : Le langage PHP	55
7.3.2 : Framework	56
7.3.3 : Choix du framework backend	56
7.3.3.1 : Présentation de Symfony	57
7.3.3.2 : Structure MVC	58
7.4 : Combinaison de ReactJS avec Symfony	59
7.4.1 : Webpack Encore	59
7.4.2 : Axios	60
7.5 : Système de gestion de base de données SGBD	60
7.5.1 : . Le choix du SGBD	60
7.5.2 : MySQL	61
7.5.3 : XAMPP	61
7.5.3.1 : Serveur web	62
7.5.3.1.1 : Apache	62
7.5.4 : Outil de conception	62
7.5.4.1 : Visual Paradigm	63
7.5.4.2 : StarUML	63

7.5.5 : Choix d'outil de conception	63
7.5.6 : Environnement de développement	64
7.5.6.1 : Visual studio code	64
Chapitre 8 : MISE EN ŒUVRE ET IMPLEMENTATION.....	65
8.1 : Architecture logicielle.....	65
8.2 : Architecture matérielle.....	66
8.3 : Présentation des différentes parties de l'application web	66
8.3.1 : Enquêteur.....	66
8.3.2 : Administrateur	66
8.4 : Conception de l'IHM	66
8.4.1 : Création de menus et interfaces.....	67
8.4.2 : . Fonctionnalité.....	67
Chapitre 9 : PRESENTATION DE L'APPLICATION DEVELOPPEE	68
9.1 : Page d'authentification :	68
9.2 : COTE ADMINISTRATEUR	69
9.2.1 : Page du tableau de bord :.....	69
9.2.2 : Liste de l'Utilisateur :	69
9.2.3 : Liste des Incidents :	70
9.2.4 : Diagramme de l'Analyse :	71
9.2.5 : Liste des supports :	71
9.3 : COTE UTILISATEUR	72
9.3.1 : Liste des incidents du côté utilisateur :	72
9.3.2 : Listes de support du côté utilisateur :	72
CONCLUSION GENERALE ET PERSPECTIVE.....	XV
BIBLIOGRAPHIE.....	XVI
ANNEXE.....	XVIII
RESUME	XXI
ABSTRACT.....	XXI

INTRODUCTION GENERALE

Les étudiants en deuxième année et troisième année à l'Ecole de Management et d'Innovation Technologique (EMIT) devront faire un stage pour consolider les connaissances théoriques acquises durant l'année universitaire. Nous avons passé les trois mois de stage au sein de la Direction Générale des Impôts (DGI) à Fianarantsoa.

Actuellement, la technologie est l'une des choses les plus importants et plus évoluée de nos jours. La révolution informatique balaye le monde et change l'image de la société. C'est aujourd'hui que naît le monde de demain, où l'ordinateur sera notre compagnon quotidien. L'ordinateur a trouvé aujourd'hui sa place à la maison et dans des bureaux. Face à l'allure qu'elle prend actuellement, l'informatique est considéré comme un vecteur principal du développement de notre grande île. La meilleure solution pour s'échapper à ce dit embarras est donc l'informatisation des traitements des dossiers. Non seulement la rapidité qu'elle nous a permet aussi la facilité de manipulation des données et nous exclut de l'erreur.

Justement nous avons eu la chance d'y avoir effectué un stage dans la Direction Régionales des Impôts (DRI) en vue d'élargir les connaissances acquises au cours de quelque apprentissage. En effet, la DRI voulait informatiser la procédure de Développement d'une application permettant la mise en forme et l'intégration en base de données liée à l'exploitation du télé déclaration de l'Impôt Synthétique (IS), afin d'avoir un outil consiste à retrouver un état opérationnel après l'interruption avec un impact négatif aussi minime que possible, et nous a ainsi confié cette tâche.

Alors ce travail est la « Conception et la réalisation d'une application de traitement des incidents et demandes liée à l'exploitation du télé déclaration de l'IS » au sein de la DRI à Fianarantsoa. Ce rapport est divisé en trois parties : la première partie concerne la présentation générale, où nous pouvons voir la présentation de l'Ecole de Management et d'Innovation Technologique (EMIT), de la Direction Régionale des Impôts (DRI). La deuxième partie est consacrée à l'analyse et la conception du projet, où nous pouvons voir les méthodes utilisées. Et la dernière partie porte sur la réalisation du projet, qui est constitué de la spécification des outils de réalisation, de la mise en œuvre et de la présentation développée.

PARTIE I : PRESENTATION GENERALE

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

L'Ecole de Management et d'Innovation Technologique (EMIT) est une grande école publique au sein de l'Université de Fianarantsoa, qui a basculé totalement vers le système L.M.D. L'EMIT nous a reconnu comme étudiants.

1.1 : Historique

L'EMIT est une école universitaire publique pluridisciplinaire, rattachée à l'Université de Fianarantsoa. La grande maturité au niveau de l'enseignement et la compétence des étudiants sortants de l'établissement ont permis aux dirigeants sous l'approbation du Ministère de mettre en place la conversion du Centre en Ecole au sein de l'Université de Fianarantsoa par le Décret N°2016-1394 du 15 Novembre 2016. L'EMIT prépare d'une part le diplôme de Master en deux mentions selon trois parcours et d'autre part le diplôme de Licence en trois mentions selon cinq parcours.

Auparavant, elle était connue sous le nom du Centre Universitaire de Formation Professionnalisant (CUFP), créée par le Décret N°2005-205 du 26 Avril 2005 et dispensait le diplôme de Licence professionnelle en Administration ainsi qu'en Informatique. Mais avant cela, elle était connue également sous le nom du Centre de Formation Continue (CFC), créé par l'Arrêté Rectoral N°99-23/UF/R du 10 Mars 1999 qui formait de diplôme de Technicien Supérieur.

L'EMIT a été sélectionnée « Meilleur Etablissement » pendant le Salon de la Recherche organisé par l'Organisation Internationale du Travail les 5 et 6 Juillet 2017. Depuis l'année universitaire 2013-2014, l'école est basculée totalement vers le système Licence, Master et Doctorat (LMD). Toutes les offres de formation dispensée à l'EMIT sont habilitées par le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique. L'école propose huit parcours répartis en trois mentions, représentés sur le tableau 1.1 :

TABLEAU 1.1: TABLEAU RECAPITULATIF

Cycle	Mention Management	Mention Informatique	Mention Relations Publiques et Multimédia
Licence	Administration Economique et Sociale	Développement d'Application Internet/Intranet	Communication Multimédia
		Conception, Intégration et Gestion des Systèmes d'Information	Relations Publiques et Communication Organisationnelle
Master	Management Décisionnel	Système d'Information, Géomatique et Décision	Relations Publiques et Multimédia
		Modélisation et Ingénierie Informatique	

Source : EMIT

1.2 : Missions

L'école a pour mission, d'abord de dispenser des formations initiales et continues en informatique, en administration et en relations publiques et multimédia. Ensuite, elle offre des services connexes à l'informatique. Puis, elle forme des techniciens opérationnels immédiatement au sein des entreprises. Enfin, elle assure le perfectionnement professionnel des étudiants, des demandeurs d'emploi, des employés et des cadres d'entreprises.

1.3 : Formations existantes

L'école présente actuellement deux cycles : Licence et Master. Chaque cycle possède plusieurs parcours assurés par un chef de mention.

1.4 : Cycle Licence

L'école possède trois (03) mentions pour cinq (05) parcours en cycle Licence. Les étudiants toutes mentions confondues doivent effectuer un voyage d'études d'insertion en entreprise pour la première année de Licence (L1), un stage de réalisation en entreprise avec un rapport de stage soutenu pour les étudiants en deuxième année de Licence (L2) et un stage de fin d'études suivi de la soutenance d'un mémoire pour les étudiants en troisième année de Licence (L3). La formation dure trois années universitaire et à la fin de la formation, les étudiants obtiennent des diplômes de Licence.

1.4.1 : Mention « Management », Parcours « Administration Economique et Sociale »

La condition d'accès en première année de Licence se fait par voie de concours pour les titulaires d'un Baccalauréat général de toutes séries (A, C, D) ou d'un Baccalauréat technique G1 et G2. A l'issue de la formation, les étudiants ont les compétences de :

- Assister le Directeur Général, le Directeur des Ressources Humaines et le Directeur Administratif et Financier ;
- Gérer les Ressources Humaines ;
- Gérer une entreprise ou un projet.

1.4.2 : Mention « Informatique », Parcours « Développement d'Application Internet Intranet » et Parcours « Conception, Intégration et Gestion des Systèmes d'Information »

La condition d'accès en première année de Licence se fait par voie de concours pour les élèves titulaires du diplôme de Baccalauréat Général Scientifique (Série C et série D), Baccalauréat Technique Professionnelle et Baccalauréat Technique Technologique (Filière Industrielle, Maintenance Automobile, Ouvrage Bois, Ouvrage Métallique, Génie Civile). A l'issue de la formation, les étudiants sont compétents en :

- Administration des bases de données ;
- Administration des réseaux et systèmes informatiques ;
- Développement d'application client/serveur.

1.4.3 : Mention « Relations Publiques et Multimédia », Parcours « Communication Multimédia » et « Relations Publiques et Communication Organisationnelle »

La condition d'accès en première année de Licence se fait par voie de concours pour les titulaires d'un Baccalauréat général de toutes séries (A, C, D) ou d'un Baccalauréat technique G1 et G2. A l'issue de la formation, les étudiants ont les compétences de :

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

1.5 : Cycle Master

L'Ecole de Management et d'Innovation Technologique (EMIT) possède deux (02) mentions pour trois (03) parcours en Master Professionnel ainsi qu'en Master Recherche. La durée de la formation est de quatre semestres c'est-à-dire deux années universitaires.

1.5.1 : Mention « Management », Parcours « Management Décisionnel »

La condition d'accès en première année de Master (M1) en S7 se fait par sélection de dossier après l'obtention du diplôme de Licence en Administration Economique et Sociale, en Gestion ou en Economie.

La mention « Management » propose un parcours « Management Décisionnel » ayant pour objectif de former et d'équiper les apprenants à la maîtrise des outils d'aide à la décision 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 pour mettre à jour les connaissances de l'apprenant par la formulation de programmes de cours qui tiennent compte de ces nouveautés. Ainsi, les objectifs principaux peuvent se résumer à former des acteurs de haut niveau en management décisionnel, de préparer des cadres capables de gérer et de créer un projet de développement économique régional et national.

1.5.2 : Mention « Informatique », Parcours « Système d'Information, Géomatique et Décision » et Parcours « Modélisation et Ingénierie Informatique »

La condition d'accès en première année de Master (M1) en S7 se fait par sélection de dossier après l'obtention du diplôme de Licence en Informatique et en Mathématique et Informatique pour les Sciences Sociales (MISS). Le Recrutement en S9 se fait par validation des crédits acquis.

Le parcours « Système d'Information, Géomatique et Décision » a pour objectif de donner un panorama des recherches actuelles et émergentes en terme de système d'aide à la décision. En effet, les systèmes informatiques et la géomatique sont en plein essor, par les grilles de calcul et 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 représentent un défi pour le génie logiciel qui doit fournir 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. De plus, l'effervescence

des techniques en géomatique qui fournissent des données spatiales et temporelles dans différents domaines représente des moyens efficaces pour prendre les bonnes décisions.

1.6 : Organigramme de l'EMIT

La structure hiérarchique au sein de l'EMIT se compose d'un conseil (scientifique ou d'établissement), une direction, un collège des enseignants, des chefs de mentions et des services présents au sein de l'école. Cette structure est représentée de manière générale sur la figure 1.1

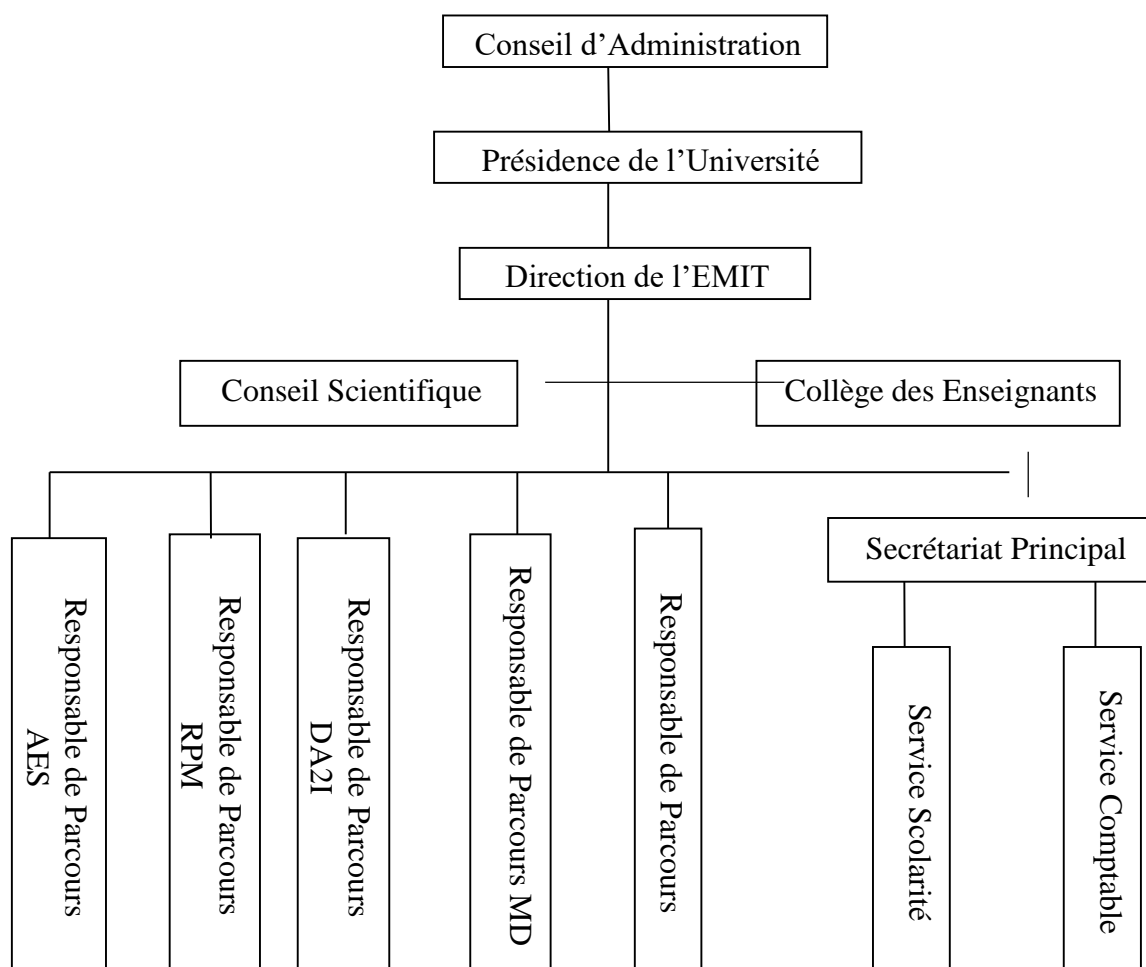


FIGURE 1.1: ORGANIGRAMME DE L'EMIT

SOURCE : EMIT

Chapitre 2 : Présentation de la Direction Régionale des Impôts (DRI)

2.1 : DRI/HM

Selon la Constitution révisée avec le retrait des ex-provinces et l'effectivité des Régions, le Ministère a changé le statut de la Direction au niveau des Provinces en Direction Régionale, c'est-à-dire que la Direction Provinciale des Impôts (DPI) devient la Direction Régionale des Impôts Haute-Matsiatra (DRI/HM) et la Direction Régionale des Impôts Vatovavy Fitovinany.

La DRI/HM a été créée (suivant décret n° 2008/242) le 25 février 2008 portant la nomination d'un Directeur Régional des Impôts de Haute-Matsiatra auprès du Ministère des Finances et du Budget.

2.1.1 : Compétences territoriales et techniques de la DRI/HM

La Direction Régionale des Impôts Haute-Matsiatra s'occupe de trois Régions : Haute-Matsiatra, Amoron'i Mania et Ihorombe. Ces trois Régions sont composées d'un Service Régional des Entreprises et comporte six Centres Fiscaux :

- Service Régional des Entreprises Fianarantsoa
- Centre Fiscal Fianarantsoa
- Centre Fiscal Ambalavao
- Centre Fiscal Ambohimahasoa
- Centre Fiscal Ambositra
- Centre Fiscal Fandriana
- Centre Fiscal Ihosy

2.1.2 : Historique du Service Régional des Entreprises Haute Matsiatra

Le Service Régional des Entreprises (SRE), ex-Service Provincial des Grandes Entreprises (SPGE), est créé par la note de service n° 519-MFB/SG/DGI du 16 juillet 2007. Il se situe au rez-de-chaussée du Bloc Administratif Tsianolondroa Fianarantsoa et y dispose 8 salles.

2.1.3 : Compétence du Service Régionales de l'Entreprises

La compétence territoriale s'étend sur les trois régions suivantes :

- Région Haute Matsiatra
- Région Amoron'i Mania
- Région Ihorombe

Territoriale technique du Service Régionale des Entreprises

En continuité des activités par l'ex-SPGE, le SRE assure la gestion, le recouvrement et le contrôle de tous les impôts, droits et taxes intérieures dus par les entreprises et les personnes physiques ou morales, réalisant des chiffres d'affaires annuels hors taxe (HT), compris entre 50 millions Ariary et 2 milliards Ariary, et relevant de sa compétence territoriale, à savoir les 3 régions citées ci-dessus. Le SRE s'occupe particulièrement des activités liées à la viticulture, du vin et des formalités spécifiques propres à ces professions : recensement des produits, opérations de prélèvements, etc.

2.1.4 : Mission de la Direction Régionale des Impôts Haute Matsiatra

Le Direction Régionale des Impôts est chargée de :

- Représenter la Direction Générale des Impôts auprès des autorités régionales et/ou locales
- Animer et coordonner les actions des services et centres fiscaux placés sous leurs ordres
- Identifier et mobiliser les moyens disponibles (financiers, matériels, humains) nécessaires pour atteindre les objectifs fixés
- Autoriser les procédures sur les produits soumis au contrôle administratif suivant les règles de compétence en matière
- Centraliser et proposer au Directeur du contrôle fiscal et du contentieux leur programme de vérification
- Organiser les programmes d'inspection des services et centres fiscaux
- Centraliser et analyser les comptes rendus de recette des services et des centres fiscaux qui en dépendent
- Transmettre les comptes rendus de recettes à la Direction de la Coordination des Opérations
- Concevoir et appliquer les mesures propres à assurer le fonctionnement rationnel des services et des centres fiscaux qui en dépendent
- Appuyer les collectivités territoriales décentralisées en matière de fiscalité.

2.1.5 : Organigramme de la Direction Régionale des Impôts Haute Matsiatra

Nous allons voir dans la figure 2.1 l'organigramme de la Direction Régionale des Impôts Haute Matsiatra

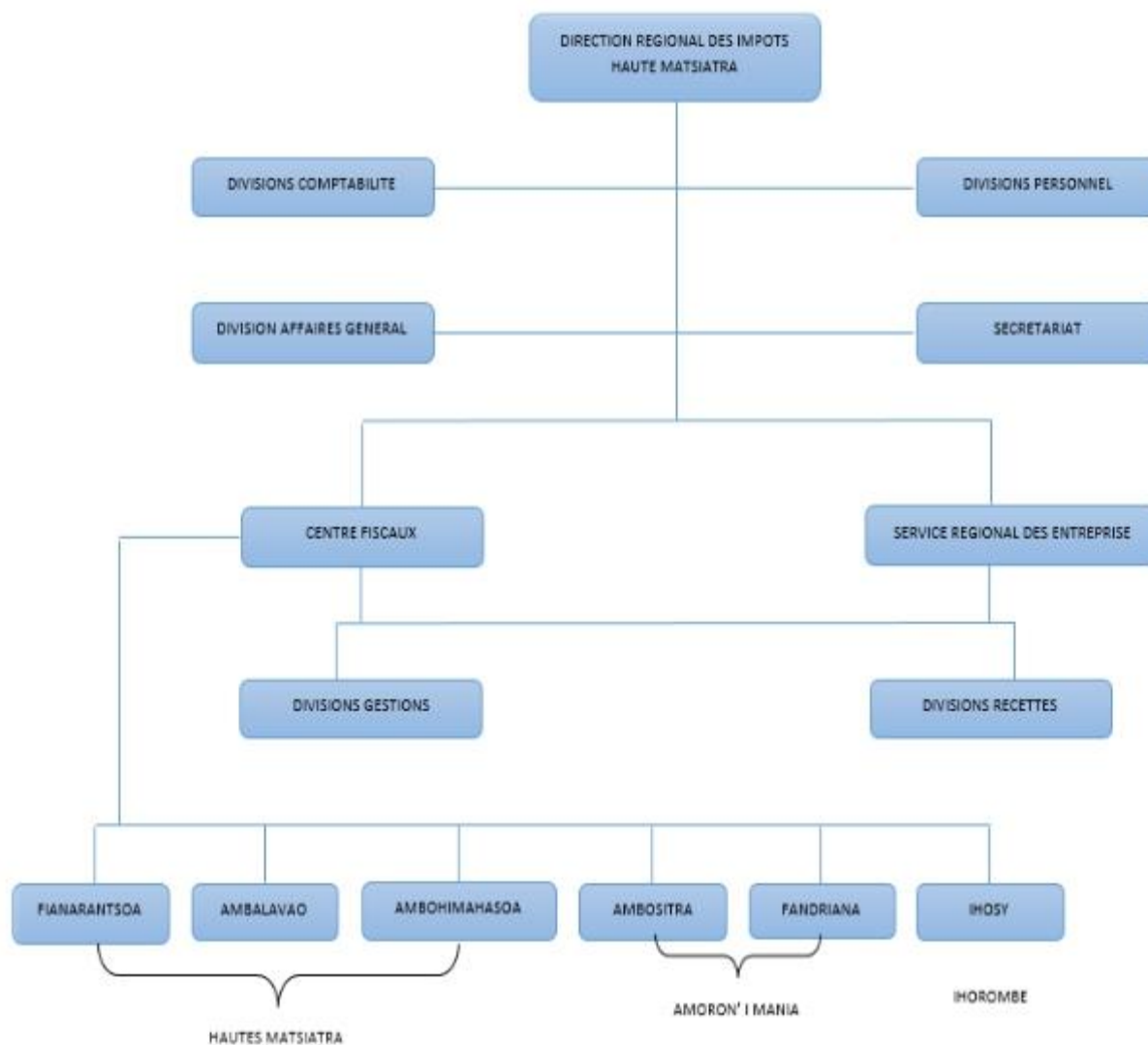


FIGURE 2.1: ORGANIGRAMME DU DRI\HM

Organisation du personnel au sein de la DRI/HM

Les personnels de la DRI/HM sont constitués des plusieurs agents dont les missions sont attribuées comme suit :

- **La division personnelle**

La division personnelle est chargée de la gestion des ressources humaines (présence, congé, etc.), elle assure la préparation, le suivi de tous les dossiers d'avancement des notations du personnel, elle est l'interlocuteur privilégié du service central chargé de la gestion du personnel pour tout ce qui concerne les dossiers administratifs du personnel de la DRI/HM, elle veille à la stricte application des décisions prises par le Directeur en ce qui concerne les avantages octroyés aux agents.

- **La division des Affaires générales**

La division des Affaires générales sont chargées d'élaborer le programme de vérification sur place, animer, coordonner et superviser les actions des vérificateurs placés sous les ordres du DRI. Elle enregistre des recettes des Centres Fiscaux et effectue le suivi et contrôle du tableau de bord mensuel venant des centres fiscaux, elle centralise les données statistiques venant au SRE et du CF.

- **L'informaticien**

L'informaticien est chargé de la maintenance et de la conception des applications Informatiques, la mise en œuvre de la sécurité informatique, assurer le bon fonctionnement de l'Intranet, et la réparation des matériels informatiques en cas de panne.

- **Le (ou la) secrétaire**

Il (ou elle) assure tous les travaux de secrétariat au niveau de son service c'est-à-dire tous les travaux de saisie, photocopie, envoi et réception. Il (ou elle) sera entre autres chargé de la gestion (réception, enregistrement, envoi) du courrier et de la tenue des divers registres (départ, arrivée), gestion des matériels mis à la disposition de son service.

- **Le dépositaire comptable La Division Comptable**

La Division comptable s'occupe de la gestion de la comptabilité des crédits et de la comptabilité matière.

- **Comptabilité des crédits**

La DRI est le gestionnaire de crédit. La comptabilité est assurée par un agent en tant que comptable.

- **Le Comptable**

Il recueille l'état C3 (budget de fonctionnement), élabore et prépare le planning de travail annuel avec le DRI. Il propose des dépenses au DRI en vue des taux de régularisation. Il prépare la certification des dossiers des fournisseurs en fonction des recommandations pour signature du

DRI, établit des factures pro-forma et taux de régularisation. Il remplit les fiches des ressources et celles des stocks pour signature du DRI et établit la répartition des fournitures de bureau pour chaque Centre Fiscal.

Il assure la réception et l'envoi des matériels à la Direction de la coordination des opérations, par exemple : matériels et mobiliers, bon de détenteurs ou ordre de sortie. Il prend en charge les matériels et autres dans le grand livre de comptabilité matière. Il établit le bon de sortie de ses matériels au détenteur effectif (ex : CF, SRE, DRI). A la fin de l'année, c'est la gestionnaire de stock qui rassemble les procès-verbaux de recensement et d'évaluation des matériels, mobiliers de bureau, matériel informatique et roulant. Par la suite, le Quitus doit être pris auprès du service Régional de Patrimoine de l'Etat Fianarantsoa qui est en principe délivré par le MFB, par la Délégation et le Directeur Régional du Budget.

- Le Chef de centre

Il programme et organise les actions des divisions placées sous ses ordres. Puis Supervise, contrôle, et dirige l'exécution des actions. Il gère les ressources humaines en service au sein du centre fiscal, centralise et analyse les réalisations.

- Le gestionnaire des dossiers Fiscaux

Il est chargé de la mise à jour des dossiers fiscaux, de la relance et du suivi des défaillances en collaboration avec la recette, l'étude sur pièces des DFU (Dossier Fiscal Unifié) pouvant aboutir à un contrôle sur pièces ou à une proposition de vérification sur place à transmettre à la division contrôle.

- Le planton

Le planton est chargé d'assurer l'envoi des diverses correspondances adressées aux contribuables, d'effectuer le dispatching du courrier parvenu chaque jour, d'assurer le nettoyage de l'ensemble des locaux, d'assurer la surveillance et la sécurité pendant la journée.

- Le chauffeur

Le chauffeur est chargé d'assurer tous les déplacements professionnels de la direction, entretien et la tenue du carnet d'entretien des véhicules de la direction et les courses urgents au centre.

2.2 : Centre Fiscale Haute Matsiatra

2.2.1 : Organigramme de CF

L'organigramme du Centre Fiscale Haute Matsiatra est représenté par la figure 2.2.

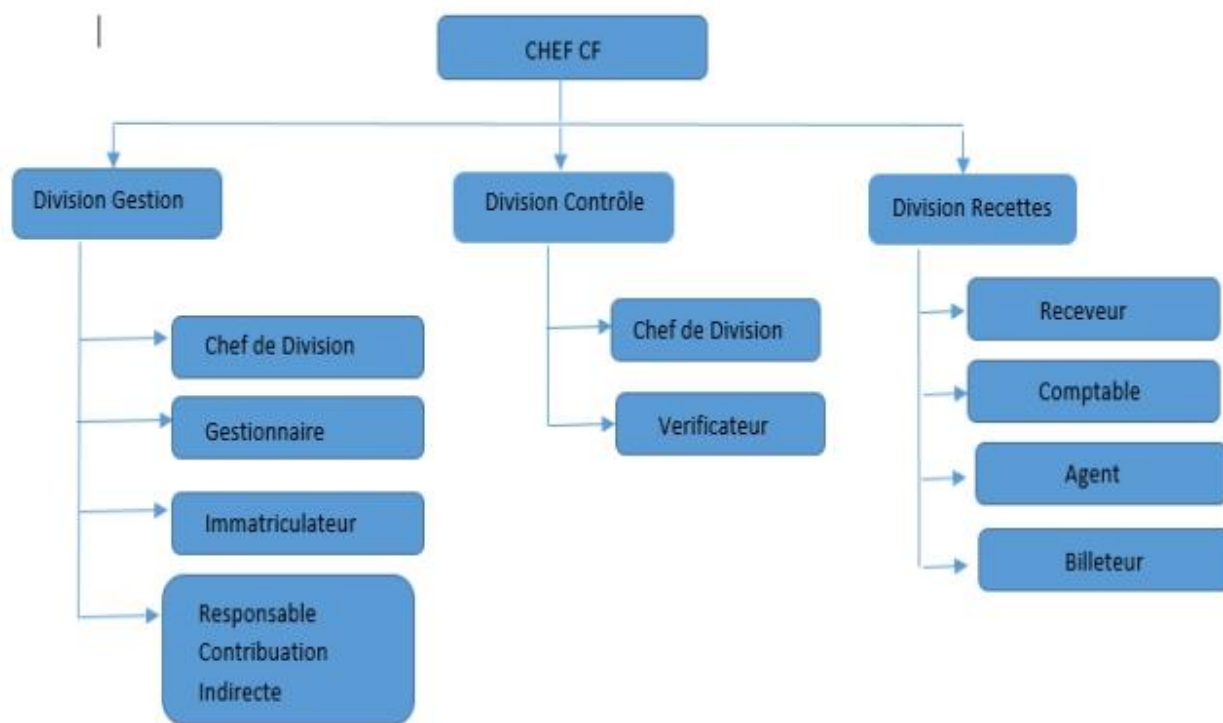


FIGURE 2.2: ORGANIGRAMME DU CF/HM

2.2.2 : Compétence territoriale

Le Centre Fiscal (CF), gère les contribuables dont les chiffres d'affaires annuels hors taxe ne dépassent pas 200 millions d'Ariary dans sa compétence territoriale :

- Le Centre Fiscal Fianarantsoa a compétence dans cinq Districts : Fianarantsoa, Vohibato, Lalangina, Isandra, Ikalamavony.
- Le centre Fiscal Ambalavao s'occupe du seul district d'Ambalavao
- Le Centre Fiscal Ambohimahasoa s'occupe du district d'Ambohimahasoa
- Le Centre Fiscal Ihosy a compétence dans trois districts : Ihosy, Iakora et Ivohibe.
- Le Centre Fiscal Ambositra a compétence dans les trois districts : Ambositra, Ambatofinandrahana, Manandriana.
- Le Centre Fiscal Fandriana s'occupe du seul district de Fandriana.

2.2.3 : Technique Fiscal

L'application de l'impôt nécessite d'en expliciter le mécanisme, c'est-à-dire de définir :

- Son champ d'application ;
- Son assiette ;
- L'exigibilité ;
- Les règles de calcul de l'impôt ;
- Les modalités de son recouvrement.

2.2.4 : Organisation du personnel au sein de la CF/HM

- Le Chef de Centre Fiscal

Il assure la gestion du personnel, et l'organisation des matériels. Il anime et coordonne toutes les actions du service général.

- Le Chef de division Gestion

Le chef de Division de Gestion des déclarations est chargé d'assurer un meilleur suivi des dossiers gérés sous sa responsabilité. Il supervise et coordonne les actions relatives aux contrôles sur pièces. Il répartit les tâches et attributions incombant aux gestions de dossiers. Il établit les comptes rendus mensuels de gestion.

- Chef de Division recette

Le chef de Division recette effectue la gestion, le contrôle des recettes est le suivi des demandes de paiements échelonnés, la gestion des dossiers contentieux et les actions en recouvrement de Procès-verbaux établis par les vérificateurs, la tenue et l'émission des comptabilités : décadaire, mensuelle et annuelle.

- Il réalise la tenue des statistiques mensuelles et des comparaisons périodiques des réalisations par rapport aux prévisions établies en début d'exercice.

- Le vérificateur

Le vérificateur est chargé de déceler les erreurs, omissions ou insuffisances dans les déclarations souscrites, vérifié par un examen approfondi de la comptabilité de l'entreprise en allant sur place recueillir des informations non contenues dans les déclarations souscrites.

- Le caissier

Le caissier est chargé de la tenue et gestion de la caisse. Il est entre autres chargé de réceptionner les contribuables, tenir le livre journal de caisse, confronter les quittances avec le livre journal de caisse, verser les espèces et chèques à la banque centrale, sortir les situations journalières décadaires et mensuelles.

- L'agent de recouvrement

L'agent de recouvrement vérifie et contrôle sur machines et sur registres, les arrières de chaque contribuable par NIF, par période et par nature d'impôts, fait la relance des mises en demeure (M.D), titre de perception (T.P), avis à tiers détenteur (A.T.D) à défaut de paiement par les contribuables, saisit et délivre le « quitus TVA »

- L'agent de recouvrement TVA, IS, DA, IR LICENCE, REDEVENCE

Il est chargé d'aider les contribuables à remplir le bordereau de versement, de consigner le cahier de versement et de perception des impôts dus, établir la quittance, le titre de liquidation pour la licence, inscrire dans le cahier de versement, la perception des impôts dus, envoyer les bordereaux de versement aux gestionnaires de dossiers, tenir la comptabilité journalière, mensuelle et annuelle, envoyer systématiquement par mois l'ampliation blanche au Receveur Général, mettre à jour le Compte Ouvert et classer par Firaïsana l'ampliation rose et les dossiers.

Chapitre 3 : Présentation du projet

Un projet est un processus unique qui consiste en un ensemble d'activités coordonnées et maîtrisées, comportant des dates de début et de fin, entreprise dans le but d'atteindre un objectif conforme à des exigences spécifiques, telles que des contraintes de délais, de confis, de coûts et de ressources.

3.1 : Contexte

Ce projet consiste à développer une application web avec une base de données portant sur le thème « Traitement des incidents et demandes liée à l'exploitation du télé-déclaration de l'Impôt Synthétique », afin de permettre la mise en forme et l'intégration en base de données liée à l'exploitation du télé-déclaration IS.

3.2 : Présentation de Traitement des incidents et demandes liée à l'exploitation du télé-déclaration IS :

3.2.1 : Définition

Le Traitement des incidents et demandes liée à l'exploitation du télé-déclaration de l'IS concerne la gestion des problèmes et des questions qui surviennent lors de la déclaration et de l'impôt par voie électronique. Il s'agit de fournir des solutions aux problèmes rencontrés par les contribuables lors de la déclaration de l'IS, tels que des erreurs de saisie, des problèmes techniques ou des questions sur la légalisation en vigueur.

3.2.2 : Les différentes types de problèmes ou des incidents peuvent survenir

Le but de notre projet est de fournir un support technique et d'aide aux contribuables pour résoudre les problèmes rencontrés lors de la déclaration de l'impôt par internet et de fournir des informations sur les différentes options de télé-déclaration disponibles. Il y a quelques problèmes peuvent survenir comme :

-Erreurs de saisie : Ces problèmes surviennent lorsque les utilisateurs entrent des données incorrectes ou incomplètes lors de la déclaration de l'IS (Ex : une erreur de saisie pourrait consister à entrer un montant d'imposition incorrect ou à ne pas fournir toutes les informations nécessaires).

-Problèmes technique : Ces problèmes surviennent lorsqu'il y a une panne ou une interruption du système de télé-déclaration de l'IS (Ex : les utilisateurs pourraient rencontrer des problèmes techniques tels que des pannes de serveur ou des problèmes de connexion internet).

-Question sur la légalisation en vigueur : ils surviennent lorsque les contribuables ont des questions sur la légalisation en vigueur concernant le télé-déclaration de l'IS (Ex : ils pourraient

avoir des questions sur les règles d'imposition, des déductions autorisées ou les déclarations obligatoires.

3.3 : Objectifs

Etant stagiaire au sein de la DGI durant ces trois mois dans le service de l'Impôt Synthétique, il est important pour nous de définir ce que fait notre service.

Ce service est d'assurer une gestion efficace des télé-déclaration de l'IS, afin de garantir un traitement rapide et fiable des problèmes rencontrés par les utilisateurs. Il s'agit notamment de résoudre les difficultés techniques, de répondre aux questions des contribuables et des professionnels, et de garantir la conformité des déclarations fiscales réalisées via cet outil.

3.4 : Origine du projet :

« Traitements des incidents et demandes liée à l'exploitation du télé-déclaration de l'impôt synthétique » trouve ses racines dans la modernisation des services publics et l'adoption croissante des technologies numériques dans la gestion des affaires fiscales. En particulier, la télé-déclaration de l'impôt synthétique (IS) a été mise en place pour simplifier et faciliter le processus de déclaration fiscale pour les petites entreprises et les contribuables individuels, en leur permettant de déclarer physique au services fiscaux.

Cependant, avec cette transformation numérique, des nouveaux défis apparaissent. Les utilisateurs peuvent rencontrer divers incidents techniques (pannes, erreurs de connexion, bugs), ainsi que des besoins d'assistance pour comprendre ou modifier leurs déclarations. Cela génère des demandes fréquentes de support et une nécessité de traiter les incidents rapidement pour assurer la continuité du service et la satisfaction des usagers. Ce thème vise donc à aborder la gestion proactive et efficace de ces incidents et demandes, pour garantir une expérience utilisateur optimale et soutenir l'adoption des services numériques dans le domaine fiscal.

L'initiative s'inscrit également dans un contexte plus large de digitalisation des administrations publiques, avec des objectifs de transparences, de réduction des coûts opérationnels, et d'accessibilité accrue aux services pour tous les contribuables, en particulier dans des régions comme Fianarantsoa où l'accès aux services peut être un défi.

3.5 : Problématique du projet :

Avec la mise en place de la télé-déclaration de l'impôt Synthétique (IS), les contribuables de Fianarantsoa bénéficient d'une plateforme numérique visant à simplifier et moderniser leurs démarches fiscales. Cependant, l'exploitation de ce système engendre des défis majeurs, notamment des incidents techniques (erreurs système, problèmes d'accès) et des

demandes d'assistance (corrections de données, navigation). Ces problèmes perturbent la continuité du service, ralentissent le traitement fiscal et impactent la satisfaction des usagers.

Comment concevoir et mettre en œuvre un système efficace de traitement des incidents et demandes, permettant d'assurer une résolution rapide, une meilleure gestion des interactions utilisateur, tout en soutenant les objectifs de modernisation et de transparence de l'administration fiscale à Fianarantsoa ?

3.6 : Planning du stage :

La planification est une étape très importante qui permet d'assurer le bon déroulement du projet tout au long des phases constituant le cycle de développement.

Il est nécessaire de définir les tâches à réaliser, maîtriser les risques et avoir une vue globale de l'avancement du projet.

Le tableau 3.1 représente le planning du stage

TABLEAU 3.1: PLANNING DU STAGE

Semaine 1-2 :	<p><u>PHASE DE PREPARATION ET COMPREHENSION DU CONTEXTE</u></p> <ul style="list-style-type: none"> -Prise de connaissance du cadre fiscal et des objectifs de la télé-déclaration. -Analyse des outils existants pour la gestion des incidents et demandes. -Entretien avec les parties prenantes (contribuables, administrateurs fiscaux). -Rédaction du cahier des charges et définition des besoins.
Semaine 3-4	<p><u>ETUDE TECHNIQUE ET CONCEPTION DU SYSTEME</u></p> <ul style="list-style-type: none"> -Recherche des solutions techniques adaptées (système de tickets, bases de données). -Modélisation du processus de traitement des incidents et des demandes. -Réalisation des diagrammes UML (cas d'utilisation, séquence, etc.). -Validation de l'architecture technique avec le tuteur de stage.
	<p><u>DEVELOPPEMENT DE L'APPLICATION</u></p>

Semaine 5-8	<p>-Développement des modules de gestion des incidents et demandes :</p> <ul style="list-style-type: none"> *Interface utilisateur pour la soumission des incidents/ demandes *Système de priorisation et de suivi des incidents. *Notification et reporting. <p>-Tests unitaires sur chaque fonctionnalité développée.</p> <p>-intégration des composants et mise en place d'une base de données.</p>
Semaine 9	<p style="text-align: center;"><u>TESTS ET VALIDATION</u></p> <p>-Tests fonctionnels et de performance de l'application.</p> <p>-Résolution des bugs et optimisation du code.</p> <p>-Validation avec un panel d'utilisateurs (administrateurs fiscaux et contribuables).</p>
Semaine 10-11	<p style="text-align: center;">DOCUMENTATION ET FORMATION</p> <p>-Rédaction de la documentation technique et utilisateur.</p> <p>-Formation des administrateurs pour l'utilisation de l'application.</p>
Semaine 12	<p style="text-align: center;"><u>FINALISATION ET PRESENTATION</u></p> <p>-Rédaction du rapport de stage</p> <p>-Préparation de la soutenance (présentation PowerPoint, démonstration de l'application).</p> <p>-Présentation finale aux parties prenantes.</p>

PARTIE II : ANALYSE ET CONCEPTION DU PROJET

Chapitre 4 : METHODE ET NOTION UTILISEE

Avant l'analyse et la conception, il faut d'abord présenter la méthode utilisée pour obtenir un rendu réaliste qui aboutira à une base de données conforme aux attentes. Dans ce chapitre, nous allons présenter d'une part, les différents cycles de vie d'un logiciel et d'autre part, le langage de modélisation que nous avons choisi.

4.1 : Méthode utilisée

4.1.1 : Généralité sur le génie logiciel

D'après Lemagit « Le génie logiciel, l'application des principes de l'ingénierie à la conception, au développement, au test, au déploiement et à la gestion de logiciels »

Est aussi appelée génie logiciel l'ingénierie appliquée au logiciel informatique, c'est-à-dire l'activité par laquelle le code source d'un logiciel est spécifié puis produit et déployé. Le génie logiciel touche au cycle de vie des logiciels. Toutes les phases de la création d'un logiciel informatique y sont enseignées : l'analyse du besoin, l'élaboration des spécifications, la conceptualisation du mécanisme interne au logiciel ainsi que les techniques de programmation, le développement, la phase de test et finalement la maintenance.

Le génie logiciel consiste à appliquer des méthodologies, c'est-à-dire à développer et à utiliser des méthodes et des outils dans le but de produire des logiciels de qualité en respectant des contraintes de temps et de coût.

4.1.1.1 : Modèle de cycle de vie

Le « cycle de vie d'un logiciel » désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. On parle du cycle de vie d'un logiciel pour définir des repères intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés et la vérification du processus de développement. La création d'un logiciel est très complexe et nécessite un ordre précis des étapes.

Découper le cycle de vie d'un logiciel en étapes "unitaires" va permettre de mettre en place des jalons et des processus de vérification et de validation qui leur seront associés. Le cycle de vie d'un logiciel va nous permettre d'assurer la qualité du produit et maintenir le coût du projet, de maîtriser les coûts de réalisation.

4.1.1.2 : Les étapes d'un cycle de vie

Le cycle de vie du logiciel comprend généralement au minimum les activités suivantes :

- **La définition des objectifs** : Cette étape consiste à déterminer la finalité du projet et son inclusion dans une stratégie globale.
- **L'analyse des besoins, appelée aussi spécifications des besoins** : Cette phase doit contenir la mise du logiciel dans son contexte (type de produit, nouveau/altéré) et l'étude de l'existant. L'étude de l'existant désigne l'étude des produits similaires dans le marché (veille concurrentielle) et l'étude du processus ou des logiciels similaires à l'entreprise. Les besoins de l'entreprise peuvent contenir des besoins fonctionnels (des fonctionnalités que le produit doit automatiser) et des besoins non fonctionnels (disponibilité, rapidité de calcul...).
- **La conception** : La conception utilise les spécifications pour décider des solutions proposées. Elle peut contenir la description des fonctionnalités de l'application. Il s'agit des fonctionnalités précisées lors de la spécification des besoins. La conception peut contenir la conception des interfaces, la conception des données et la conception de l'architecture matérielle.
- **Le codage** : Il transforme des solutions proposées lors de la conception en un code opérationnel. Les techniques de codage dépendent du langage et doivent être bien conforme à la conception.
- **L'intégration** : Il s'agit de la phase de test et de validation. Les tests déterminent les bugs techniques, les bugs fonctionnels et la qualité du logiciel. Pour cela, on peut utiliser des logiciels de test, des techniques et des benchmarks.
- **La mise en production** : Cette étape regroupe toutes les activités qui mènent à l'installation et la mise en marche de l'application développée (installation des serveurs, setup et configuration des composants du logiciel développé et le test de déploiement).
- **La maintenance** : Cette étape consiste à ajuster l'application après la livraison du produit au client. Elle a pour but de corriger les erreurs et les anomalies du système et modifier le système pour y ajouter des fonctionnalités.

4.1.2 : Modèles de cycle de vie de développement logiciel

Pour bien organiser le processus de développement, quatre modèles ont été définis, à savoir :

- Le modèle en cascade ;
- Le modèle en V ;
- Le modèle en spirale ;
- Le modèle par incrément.

Nous allons les voir en détails un par un.

4.1.2.1 : Modèle en cascade

Considéré comme le modèle, à la fois le plus ancien et le plus simple de tous, ce modèle se divise en plusieurs phases. Chaque phase a son propre plan et commence après la précédente. Le résultat de la première phase représentera le point d'entrée de la phase d'après. Ce modèle nécessite beaucoup de documentation qu'on peut trouver facilement, à chaque phase, au niveau de la phase antérieure. Seulement il faut faire attention à un point. Tous les détails, même les plus petits, doivent être complétés. Le cas échéant, l'ensemble du processus sera retardé.

À la fin de chaque étape, des documents ou alors des logiciels sont produits. Ces résultats sont revisités de façon approfondie et on ne passe à la phase suivante que si ces résultats sont jugés satisfaisants. Après quelques modifications du modèle en cascade original, les possibilités de retour arrière ont été ajoutées. Seulement une phase ne peut remettre en cause que la phase précédente, ce qui est insuffisant du point de vue pratique. Ce modèle repose sur l'hypothèse souvent idéaliste que l'on peut dès le départ, fixer en détail et complètement ce que l'on veut réaliser. Un autre inconvénient de ce modèle est que la vérification du bon fonctionnement du système s'effectue très tardivement. La figure 4.1. Représente le type de modèle en cascade d'un logiciel.

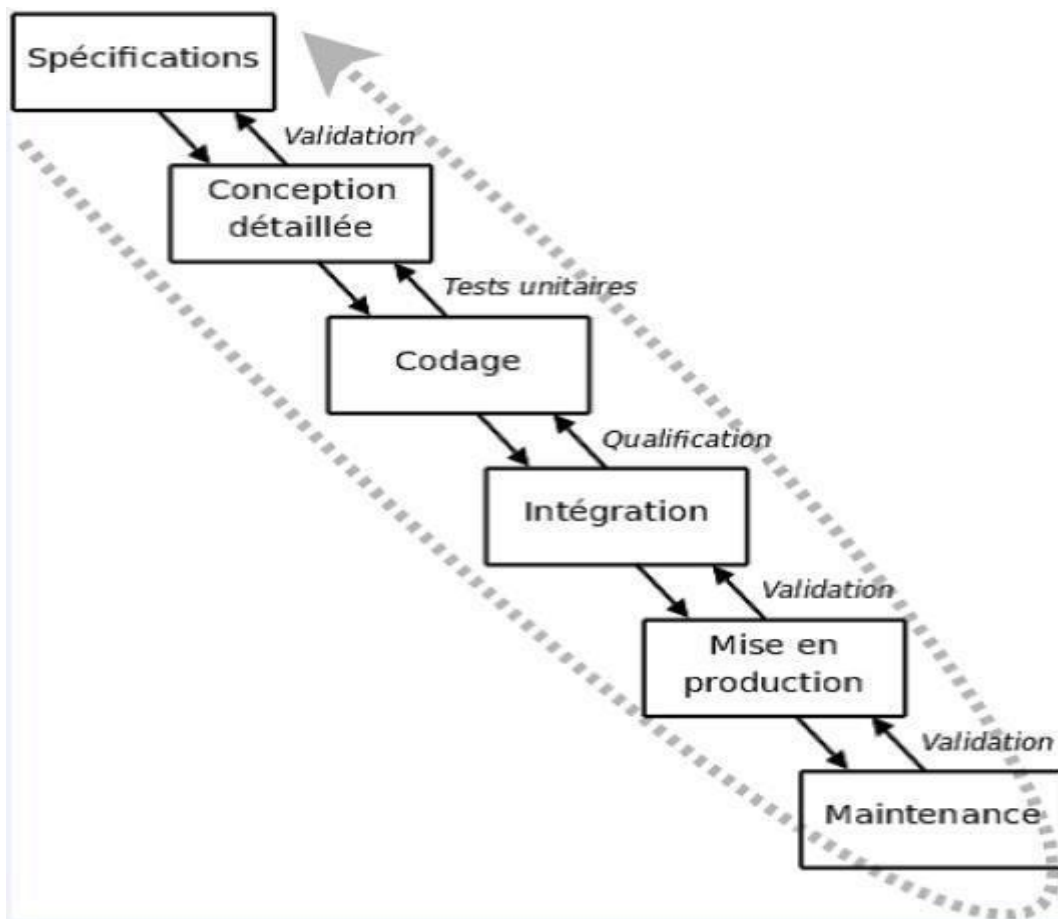


FIGURE 4.1: MODELE EN CASCADE

4.1.2.2 : Modèle en V

Dans ce type de modèle, les différentes phases sont planifiées en parallèle. Les phases de vérification du logiciel et de sa validation se font parallèlement, en simultané.

Il est représenté par un V dont la branche descendante contient toutes les étapes de la conception du projet, et la branche montante, toutes les étapes de test du projet. La pointe du V représente la réalisation concrète du projet, le codage. On pourrait donc en déduire, de manière simple, que les deux branches du V ne sont que de la documentation.

Le modèle en V est actuellement le modèle de cycle de vie le plus connu, et certainement le plus utilisé. Ce modèle consiste à reposer sur une interdépendance des étapes soumises à une validation avant la prochaine étape et une vérification anticipatoire du produit final.

La figure 4.2 représente schématiquement le modèle de cycle de vie en V.

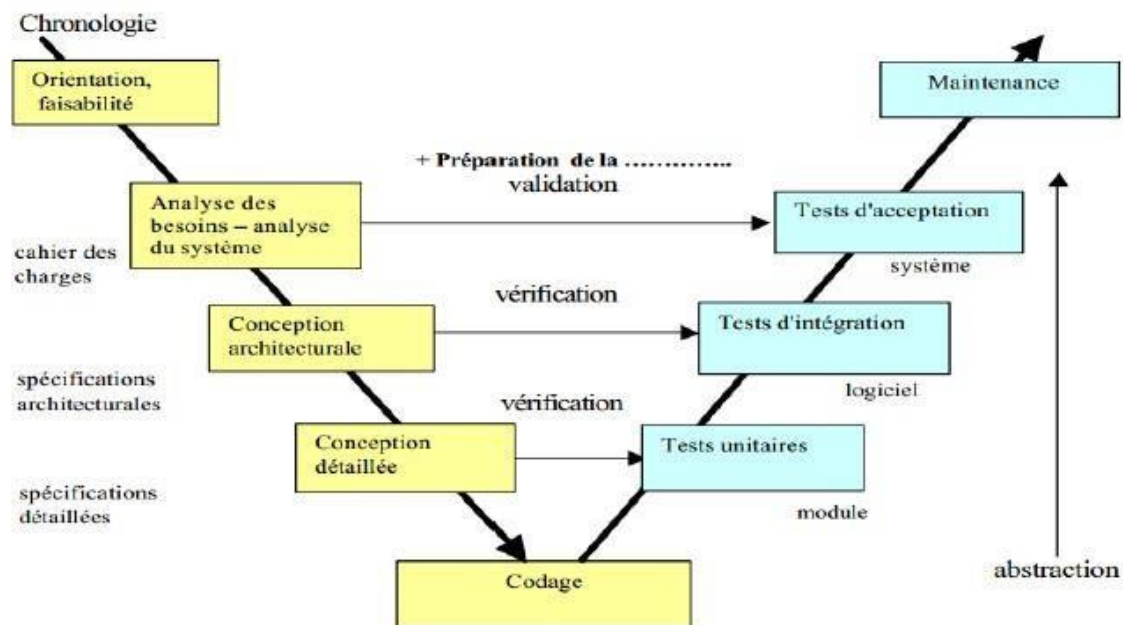


FIGURE 4.2: MODELE EN V

Modèle en spirale

Parmi les modèles de cycle de vie, c'est celui qui est le plus flexible. À chaque passage, de nouvelles améliorations sont apportées. On peut considérer que le modèle en spirale est une combinaison réussie des avantages des modèles de prototypage et celui en cascade (rapidité et simultanéité dans les activités de conception et de développement).

Le modèle en spirale est un générateur de modèle qui se base sur le risque pour des projets. Le modèle guide une équipe à adopter les éléments d'un ou plusieurs modèles de processus.

On distingue quatre (04) phases dans le déroulement du cycle :

- Détermination des objectifs, des alternatives, des contraintes ;
- Analyse des risques, évaluations des alternatives ;
- Développement et vérification de la solution retenue ;
- Revue des résultats et vérification du cycle suivant.

Le processus de développement est représenté par une spirale plutôt qu'une séquence d'activités avec retour arrière éventuels. Chaque boucle dans la spirale représente une étape du processus de développement. Les risques sont explicitement adressés et résolus tout au long du processus.

La représentation schématique du modèle de cycle de vie en spirale est montrée par la figure 4.3 suivante :

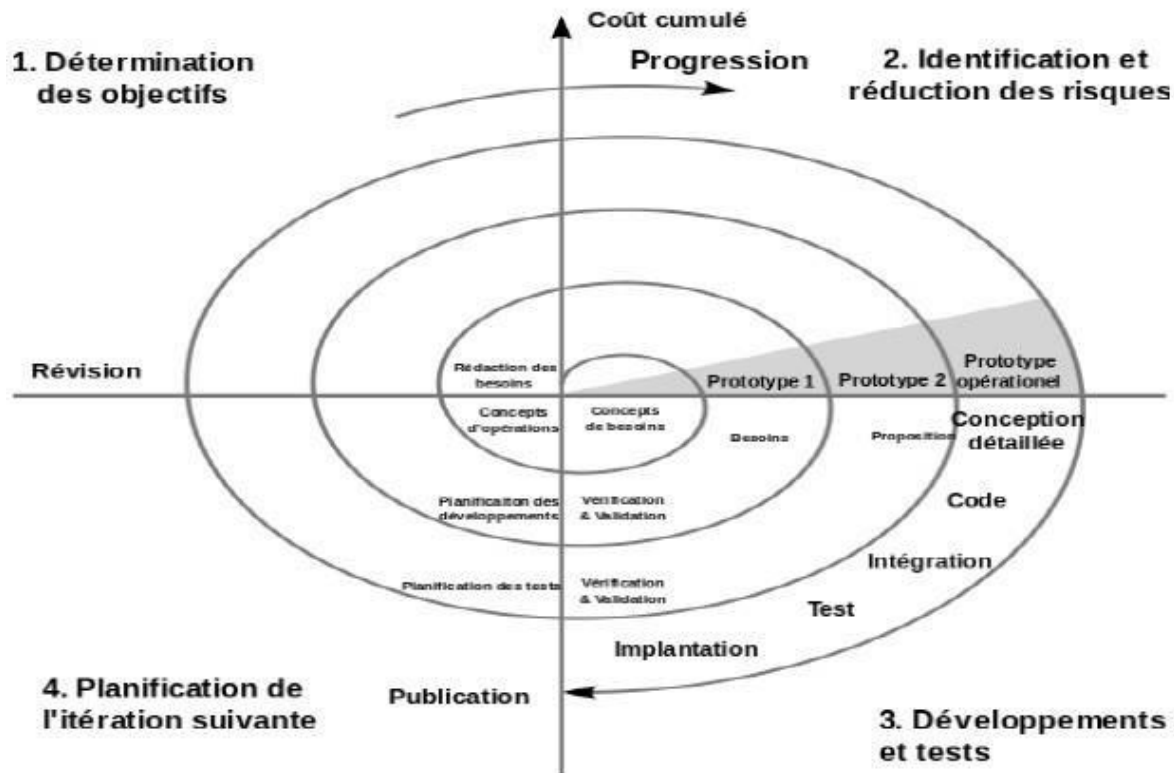


FIGURE 4.3: MODELE EN SPIRALE

4.1.2.3 : Modèle de cycle de vie par incrément

Le modèle de cycle de vie par incrément consiste à développer à la fois un seul ensemble de composant, contrairement aux modèles précédents. Il s'agit d'une intégration des incréments à un noyau de logiciel développé au préalable. Ce modèle a pour principe de diviser le projet en incréments (sous-partie fonctionnelle cohérente du produit final), chaque incrément ajoute de nouvelles fonctions et est testé comme un produit final. Ce modèle a pour avantages de moins de complexités pour chaque développement, de progressivités des intégrations, de la possibilité de livraison et de mise en service pour chaque incrément. Les risques de ce type de modèle sont la remise en cause des incréments précédents ou le noyau, et le non intégration de nouveaux incréments. La figure 4.4 représente le modèle par incrément.

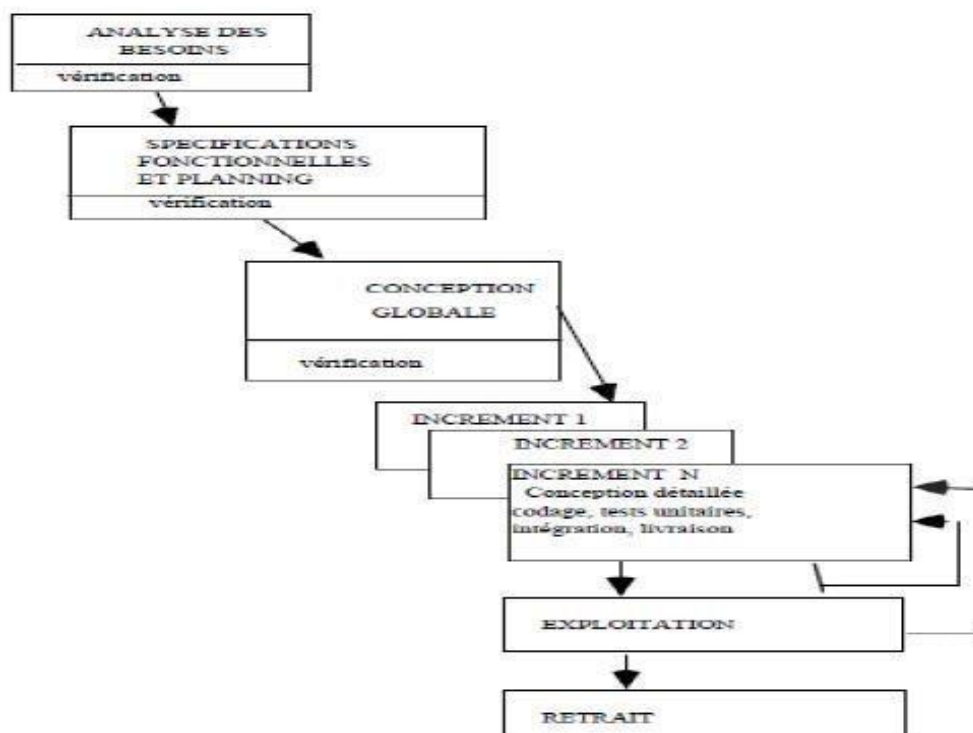


FIGURE 4.4: MODELE PAR INCREMENT

4.1.3 : Choix du modèle de cycle de vie

Chacun des modèles présentés auparavant a ses avantages et ses inconvénients. En comparant ces différents modèles, nous allons faire un choix parmi eux et utilisé le mieux adapté pour nous. Le tableau 4.1 illustre la comparaison des modèles de cycle de vie.

TABEAU 4.1: COMPARAISON DES MODELES DE CYCLE DE VIE

Modèle de cycle de vie	Avantages	Inconvénients
En Cascade	- Facile à comprendre et à utiliser ;	- Tous les besoins doivent être bien définis au départ ;
	- Adapté pour une équipe inexpérimenté ;	- Pas d'interaction entre les phases de development;
	- Les limites de chaque étape sont visibles;	- L'intégration n'a lieu qu'à la fin du cycle;
	- La définition des besoins est non évolutive.	- Le client peut se retrouver non satisfait;
		- Pas de retour en arrière d'une étape à l'autre.

<p>En V</p>	<ul style="list-style-type: none"> - Facile à utiliser; - Les tests sont effectués à chaque étape; - Le contrôle se fait progressivement à chaque étape; - Favorise la décomposition hiérarchique fonctionnelle; - Les phases de validation sont prises en main très tôt dans le processus de développement. 	<ul style="list-style-type: none"> - Une mauvaise prise en compte des événements concurrents; - Le processus n'est pas itératif; - Une mauvaise prise en compte des changements de la spécification des besoins ; - Ne contient pas d'activités d'analyse de risque.
<p>En spirale</p>	<ul style="list-style-type: none"> - Les fonctions critiques à haut risques sont développés en premier lieu; - Validation concrète et non sur documents; - Limitation du risque à chaque itération; - La conception ne doit pas forcément être achevée ; - Les utilisateurs finaux sont intimement associés à toutes les étapes de développement; - Le développement se fait avec interaction avec les clients; - Les utilisateurs ont dès le départ une vue globale du système. 	<ul style="list-style-type: none"> - Requiert la définition complète des fonctionnalités du système pour une définition des différents incréments; - Le temps consacré à l'évaluation des risques est très élevé pour de petits projets; - Ce modèle est complexe ; - Une expertise en évaluation des risques est nécessaire; -Le spirale peut être infinie; - Il est difficile de définir les -objectifs et les points de validation intermédiaire entre les différentes étapes.

<p>En incrément</p>	<ul style="list-style-type: none"> - Le client peut valider chaque étape de processus; - Utilisation de la méthode « diviser pour mieux régner »; - Le coût du lancement est moindre; - Un produit exploitable peut être délivré à tout moment; - Le risque de changement des besoins est minimal. 	<ul style="list-style-type: none"> - Requiert une bonne conception; - Le coût total du développement du système n'est pas négligeable; - Les différentes interfaces doivent être bien définies.
----------------------------	---	--

Chaque modèle de cycle de vie a ses propres points forts et points faibles. A partir de ce tableau, nous avons décidé d'utiliser le modèle de cycle de vie en V, car il est facile à utiliser, il y a des contrôles progressifs à chaque étape et favorise la décomposition hiérarchique fonctionnelle.

Malgré le fait qu'il ne contient pas d'activités d'analyse de risque, ce modèle est un modèle éprouvé par calque sur la production industrielle classique, il permet l'organisation du travail et des équipes, il propose des étapes clés (documentation, revues) qui entraîne un bon suivi du projet, il permet de garantir une certaine qualité, et enfin il est adapté à de grand projet. Il est aussi le plus efficace avec son principe de travail qui nécessite la vérification de chaque étape et la possibilité de corriger les fautes avant de se lancer vers l'étape suivante.

4.2 : Le langage Unified Modeling Language

4.2.1 : Histoire et origine de l'UML

Les Three Amigos du génie logiciel, comme on les appelait alors, avaient élaboré d'autres méthodologies. Ils se sont associés pour apporter plus de clarté aux programmeurs en créant de nouvelles normes. La collaboration entre Grady, Booch et Rumbaugh a renforcé les trois méthodes et a amélioré le produit final.

Les efforts de ces penseurs ont abouti à la publication des documents UML 0.9 et 0.91 en 1996. Il est rapidement devenu évident que des sociétés comme Microsoft, Oracle et IBM voyaient l'UML comme un élément critique pour leur développement futur. Elles ont donc mis en place des ressources, accompagnées en cela par de nombreuses autres sociétés et personnes, permettant de développer un langage de modélisation complet. Les Three Amigos ont publié

The Unified Modeling Language User Guide en 1999, qui fut suivi d'une mise à jour comportant des informations sur l'UML 2.0 en 2005.

4.2.2 : Définition

Le langage Unified Modeling Language ou UML est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Le langage de modélisation peut servir de modèle pour un projet et garantir une architecture d'information structurée ; il peut également aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour les spécialistes externes. UML est principalement utilisé dans le développement de logiciels orientés objet. Les améliorations apportées à la norme dans la version 2.0 la rendent également adaptée à la représentation des processus de gestion.

4.2.3 : Diagrammes UML

Les catégories principales "structure" et "comportement" représentent les concepts de base représentés par les diagrammes UML. Dans le groupe des diagrammes de comportement, UML spécifie

la sous-catégorie « diagrammes d'interaction ». La figure 4.5 représente les diagrammes

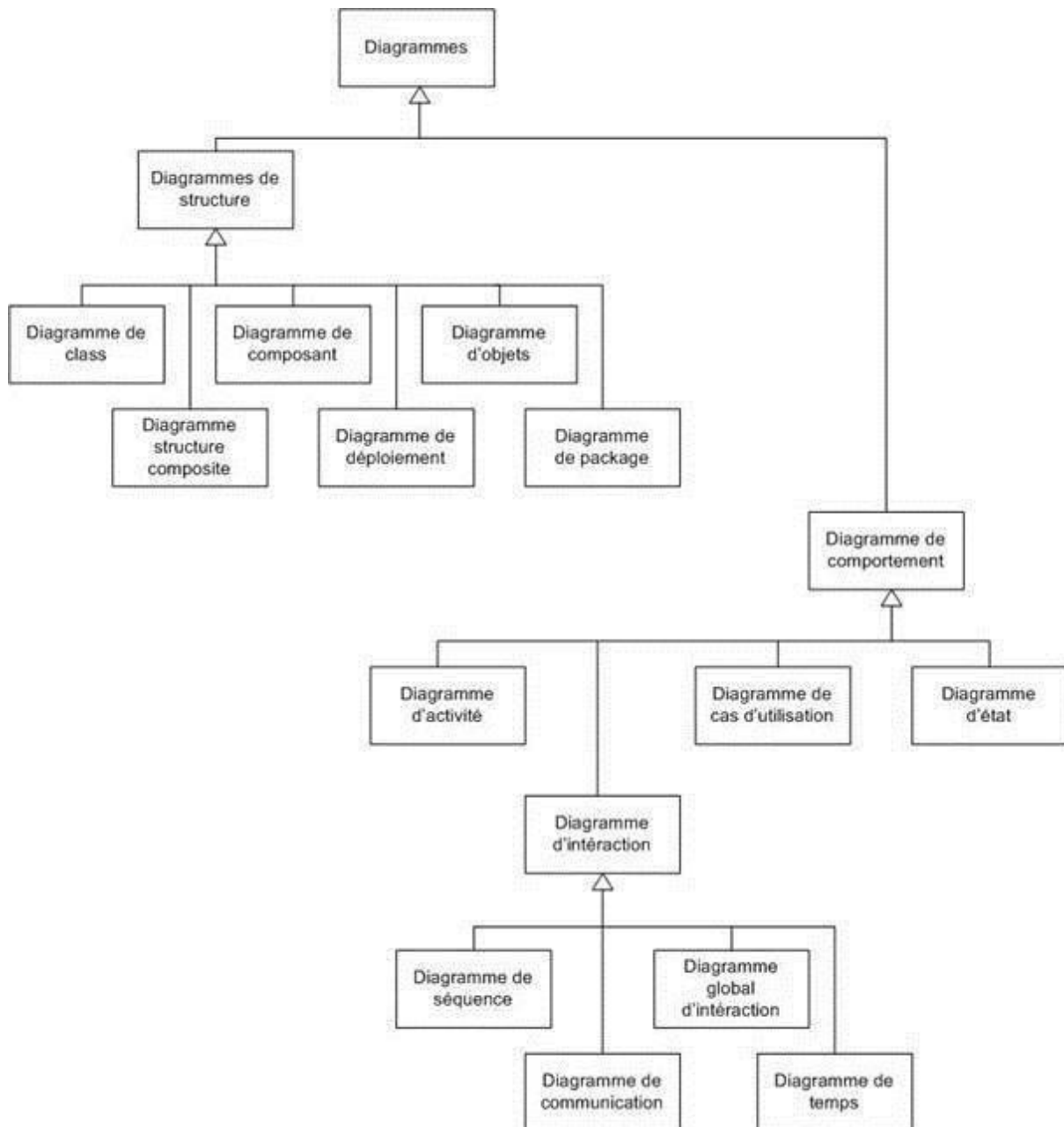


FIGURE 4.5: DIAGRAMME UML

Nous allons parler des diagrammes que nous avons utilisés dans la conception de notre application.

4.2.3.1 : . Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation 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 cas d'utilisations ou use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants à un système.

Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.

Les éléments d'un diagramme de cas d'utilisation sont :

- Acteurs qui sont les entités interagissant avec le système, il est illustré par la figure 4.6

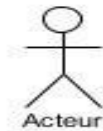


FIGURE 4.6: FIGURE DE L'ACTEUR

- Cas d'utilisations qui symbolise les différentes tâches effectuées au sein du système représenté par la figure 4.7.

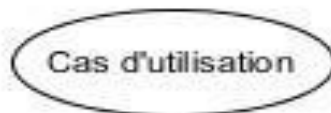


FIGURE 4.7: FIGURE REPRESENTANT LE CAS D'UTILISATION


- Relation qui exprime l'interaction existant entre un acteur et un cas d'utilisation.
- Scénario qui est une instance de cas d'utilisation.

Il existe différents types de relation que nous pouvons utiliser pour exprimer les interactions dans les diagrammes de cas d'utilisation. Le tableau 2 les représente. Le tableau 4.2 représentant les éléments du diagramme de cas d'utilisation :

TABEAU 4.2: TABLEAU REPRESENTANT LES ELEMENTS DU DIAGRAMME DE CAS D'UTILISATION

Type de relation	Représentation graphique
<p><u>Relation d'inclusion ou « include »:</u></p> <p>Un cas d'utilisation A inclus un cas d'utilisation B où le comportement décrit par A inclus le comportement de B : le cas A</p>	

Dépend du cas B

<p><u>Relation d'extension ou « extend »:</u></p> <p>Un cas d'utilisation A étend un cas d'utilisation B lorsque A peut être appelé au cours de l'exécution B</p>	
<p>Relation entre acteur et utilisation :</p> <p>Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu.</p>	

4.2.3.2 : Diagramme de séquence

Le diagramme de séquences désigne l'ordre temporel d'un cas des opérations effectuées par un acteur, il indique les objets manipulés par les acteurs et les opérations qui font passer d'un objet à un autre. Les entités communiquent entre eux à travers des messages représentés sous forme de flèches.

Pour chaque cas d'utilisation, on considère deux éventualités possibles, nommées :

- Scénario nominal : c'est la représentation d'un cas d'utilisation lorsque celui-ci se passe de la meilleure des façons ou tout simplement de manière normale ;
- Scénario alternatif : c'est la représentation d'un cas d'utilisation sous le pire scénario, elle est caractérisée par un échec de l'opération.

Le tableau 4.3 représente la description de chaque ligne du diagramme de séquence :

TABLEAU 4.3: DESCRIPTION DE CHAQUE LIGNE DU DIAGRAMME DE SEQUENCE

Ligne	Description
Cas d'utilisation	Le nom du cas d'utilisation étudié
Pré condition	Les conditions nécessaires au préalable du cas d'utilisation pour qu'il soit exécutable
Scénario nominal	Scénario normal du cas d'utilisation
Scénario alternatif	Le pire des scénarios

Post-conditions	Les conditions nécessaires à l'achèvement du cas d'utilisation
-----------------	--

4.2.3.3 : Diagramme de classes

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'information. Il permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles sont regroupées dans des classes. Le diagramme met en évidence d'éventuelles relations entre ces classes.

Le diagramme de classes comporte 6 concepts : classe, attribut, identifiant, relation, opération et Généralisation / spécialisation.

Le diagramme de classe est illustré par la figure 4.8

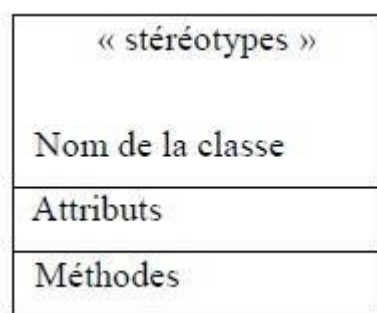


FIGURE 4.8: DIAGRAMME DE CLASSE

Les différents types de visibilité sont montrés dans le tableau 5

TABLEAU 4.4: DIAGRAMME DE CLASSE

Modificateur	Visibilité
Privé (-)	Seuls les membres de la classe uniquement
Public (+)	Accessibles par toutes les classes
Package (~)	Accessibles par toutes les classes de même package.
Protégé (#)	A l'intérieur de la classe et dans le sous classes

Le diagramme de classes en relation est représenté par la figure 4.9

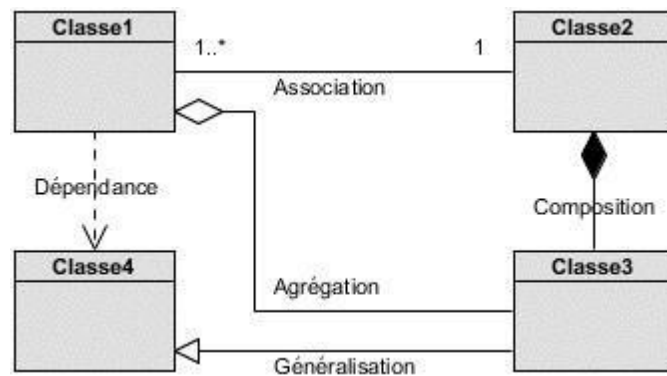
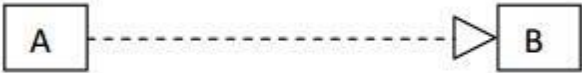


FIGURE 4.9: DIAGRAMME DE CLASSE EN RELATION

Il existe différents types de relation que nous pouvons utiliser pour exprimer les interactions dans les diagrammes de classes. Le tableau 4.5 représente les différents types de relations :

TABEAU 4.5: DIFFERENTS TYPES DE RELATIONS

Type de relation	Représentation graphique
Association : Elles représentent un lien durable ou ponctuel entre deux objets, une appartenance, ou une collaboration. L'association permet à A d'atteindre B.	
Généralisation et héritage: B est la classe de base et A est la classe dérivée (spécialisée). La classe A peut posséder toutes les caractéristiques de sa classe parent B.	
Dépendance : C'est une relation unidirectionnelle. La	

modification de B peut entraîner la modification de A.	
<p>Dépendance:</p> <p>C'est une relation unidirectionnelle.</p> <p>La modification de B peut entraîner la modification de A.</p>	
<p>Agrégation :</p> <p>Lorsqu'un objet en contient d'autres, on parle d'agrégation. A est inclus dans B.</p>	
Composition (agrégation composite) :	
Quand le composite (B) est détruit, le composant (A) l'est aussi	
<p>Implémentation:</p> <p>Une classe peut implémenter une interface.</p> <p>Les interfaces se différencient des autres</p> <p>Classes par le stéréotype <<interface>>.</p>	 <pre> classDiagram A .. > B </pre>

4.2.4 : Avantages et inconvénients d'UML

Le choix de l'UML est justifié par plusieurs facteurs que nous pouvons voir dans la colonne « Avantages ». Les avantages de l'UML sont illustrés dans le tableau 4.6

TABEAU 4.6: AVANTAGES ET INCONVENIENTS DE L'UML

Avantages	Inconvénients
<p>UML est un langage formel et normalisé</p> <ul style="list-style-type: none">- gain de précision- gage de stabilité- encourage l'utilisation d'outils <p>UML est un support de communication performant</p> <ul style="list-style-type: none">- Il cadre l'analyse.- Il facilite la compréhension de représentations abstraites complexes.- Son caractère polyvalent et sa souplesse en font un langage universel.	<ul style="list-style-type: none">- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.- L'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue

Chapitre 5 : ANALYSE DU PROJET

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'analyse des besoins, l'analyse de l'existants, étude de faisabilité, évaluation de faisabilité, les critiques de l'existants et propositions des solutions.

5.1 : Etude 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. Elle a été menée en analysant les données d'incident et de demandes précédentes, en consultant les responsables et en effectuant des recherches sur les meilleures pratiques en matière de gestion des incidents et des demandes. Cela permet d'éprouver les détails du domaine à étudier et ensuite apporter les solutions fonctionnelles et techniques

5.2 : Analyse de besoins

Les analyses des besoins identifient les problèmes rencontrés par les contribuables lors de la déclaration et de l'exploitation de l'IS, évaluent leur complexité et leur impact sur les activités de la DGI, et identifient les moyens nécessaires pour résoudre les problèmes et améliorer les services offerts. Les besoins principaux sont : Identifications des problèmes rencontrés par les contribuables lors de la déclaration et de l'exploitation de l'IS.

- Evaluation de la complexité des problèmes et de leur impact sur les activités de la DGI.
- Reconnaissances des besoins en termes de formation et d'informations des agents de la DGI.
- Identifications des moyens nécessaires pour résoudre les problèmes et améliorer les services offerts.

5.3 : Analyse de l'existants :

Prenons compte en premier lieu l'état physique des outils informatiques au sein de la CFA Fianarantsoa. Nous pouvons connaître tous les matériels et leurs caractéristiques dans le tableau ci-dessous.

TABLEAU 5.1: EXISTANT MATERIEL DE L'ENTREPRISE

Désignation	Marque	Processeur	Disque Dur	Fréquence	Mémoire RAM	Système d'exploitation
Ordinateur de bureau	HP/COMPACT	Pentium II	80 Go	350 MHz	828 Mo	Windows XP Swith 3

5.4 : Etude de faisabilité :

L'étude de faisabilité doit permettre d'évaluer les possibilités techniques, économiques et organisationnelles de mettre en œuvre des solutions pour résoudre les problèmes. Il est nécessaire avant le lancement du projet d'évaluer si le projet est fiable ou non. Cette phase nous permet de définir ce qui sera inclus dans les objectifs du projet, ce qui ne le sera pas et si le projet doit bien lancer. Il permet aussi de déterminer si l'organisation est bien en mesure de mener le projet à son terme. On cherche en particulier à savoir si elle dispose des compétences, des ressources et des fonds nécessaires.

5.5 : Evaluation de faisabilité :

5.5.1 : Faisabilité organisationnelle :

Le projet ne nécessite pas une modification de structure de l'organisation de la Direction. Les personnels n'auront besoin que de formation lors de la mise en place du logiciel.

5.5.2 : Faisabilité technique :

Les ressources matérielles existantes sont amplement suffisantes pour le développement ainsi que la mise en marche. Les techniques et les méthodes de conceptions seront similaires aux autres applications déjà opérationnelles au sein de la DGI.

5.5.3 : Faisabilité économique :

Etant donné que c'est un projet d'essai, il n'y a pas d'investissement. Ainsi, malgré les contraintes, le projet reste faisable et peut être mise en place.

5.6 : Les critiques de l'existant :

Elles peuvent se poser en trois niveaux à savoir :

Niveau informationnel, niveau organisationnel, niveau technique.

5.6.1 : Niveau informationnel :

Les informations disponibles sur le site web de la DRI à Fianarantsoa sont parfois peu claires ou difficiles à trouver, ce qui peut poser des problèmes aux utilisateurs. De plus, certains outils nécessaires à la déclaration et à la gestion de l'impôt synthétique peuvent être complexes à utiliser, notamment pour les utilisateurs non expérimentés en matière d'informatique fiscale.

5.6.2 : Niveau organisationnel :

La réponse aux demandes d'aide et aux problèmes techniques peut être lente ou inadéquate en raison de la charge de travail du service clientèle de la DRI à Fianarantsoa. De plus, le manque de coordination entre les différentes équipes de la DRI à Fianarantsoa peut entraîner des délais dans la prise en charge des demandes d'aide ou des problèmes techniques.

5.6.3 : Niveau techniques :

Le site web de la DRI à Fianarantsoa est parfois sujet à des pannes ou des erreurs de fonctionnements, ce qui peut entraîner des problèmes pour les utilisateurs lors de la déclaration et la gestion de leur impôt synthétique. De plus, certains outils techniques utilisés par la DRI à Fianarantsoa peuvent être obsolètes ou non conformes aux normes en vigueur, ce qui peut entraîner des problèmes de sécurité ou de fiabilité.

5.7 : Les solutions proposées :

Pour résoudre ces problèmes, il est proposé de mettre en place un système de suivi et de gestion des tickets. Ce système permettra de suivre l'évolution des incidents et demandes, de les classer par priorité et de les répartir aux agents compétents.

Une évaluation régulière des processus de traitement est également proposée pour améliorer la qualité du service aux contribuables. Cette évaluation pourrait être réalisée par l'utilisation des indicateurs clés de performances (KPI) tels que le temps de réponse aux incidents et aux demandes, le nombre d'erreurs commises ; etc.

Enfin, une formation continue et une sensibilisation des incidents et des demandes liées à l'exploitation du télé déclaration de l'IS sont des éléments clés pour réduire le risque de problème potentiels et garantir la pérennité du système. Des simulations d'incidents et des exercices de gestion de demandes pourraient être organisés régulièrement pour améliorer les compétences des agents.

Chapitre 6 : CONCEPTION DU PROJET

La conception du projet est la mise en œuvre d'un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus permettent la conception, l'écriture et la mise au point d'un logiciel jusqu'à sa livraison au demandeur.

Donc, entrons maintenant dans la phase de conception y compris le dictionnaire de données, les règles de gestion et ensuite les diagrammes.

6.1 : DICTIONNAIRE DES DONNEES

Le dictionnaire des données décrit la totalité de donnée manipulées ou la liste de champs pour la conception de futur système, il est représenté sous formes d'un tableau. Dans le tableau suivant, nous pouvons trouver la liste symbolisée (rubrique du dictionnaire) pour éviter l'usage de mot trop long qui se trouve à la description. La représentation de dictionnaire des données et de notre futur système est dans le tableau ci-dessous.

TABEAU 6.1: REPRESENTATION DE DICTIONNAIRE DE DONNEES

RUBRIQUE	DESCRIPTION	TYPE	LONGUEUR	ETAT
-idFichier	Identité fichier	N	5	E
-nomFichier	Nom de fichier	A	50	E
-typeFichier	Type de fichier	AN	60	E
-lienFichier	Lien de fichier	AN	50	E
-idDiscussion	Identité de discussion	N	5	E
-dateCreation	Date de création	D	20	E
-idUtilisateur	Identité de l'utilisateur	N	5	E
-nomUtilisateur	Nom de l'utilisateur	AN	60	E
-mailUtilisateur	Mail de l'utilisateur	AN	40	E
-motdepasse	Mot de passe	AN	120	E
-typeUtilisateur	Type de l'utilisateur	AN	70	E
-urlPhoto	Url du photo	AN	120	E
--idMessage	Identité du message	N	5	E
-contenuMessage	Contenu du message	AN	250	E
-dateMessage	Date de message	D	15	E
-statutMessage	Statut de message	AN	95	E

-idLog	Identité de Log	N	5	E
-contenuLog	Contenu de Log	AN	250	E
-dateLog	Date de Log	D	20	E
-idIncident	Identité de l'incident	N	5	E
-descripIncident	Description de l'incident	AN	250	E
-typeIncident	Type de l'incident	AN	50	E
-debutIncident	Date début de l'incident	D	15	E
-finIncident	Date fin de l'incident	D	15	E
statutIncident	Statut de l'incident	AN	65	E
-prioriteIncident	Priorité de l'incident	AN	80	E
-dateCreation	Date de création	D	15	E

6.1.1 : Types de dictionnaires du donnés :

- Alphabétique (A) : la donnée est uniquement composée de caractères alphabétiques.
- Numérique (N) : la donnée est composée uniquement de nombres (entiers ou réels).
- Alphabétiques Numérique (AN) : la donnée peut être composée à la fois de caractère alphabétique et numérique.
- D : Date.
- E : Elémentaire.

6.2 : Les règles de gestions :

D'après Wikipédia, En ingénierie : « Les exigences doivent être écrite de telle manière qu'elles orientent la création et la modification d'un système selon les règles métier (ou règles de gestions) appropriée au contexte et au domaine, et dans lequel le système doit être utilisée ».

Avant de modéliser le système, il est donc important de se renseigner sur les exigences qui réagissent les entités de la DRI, les règles de gestions sont la formulation de ses spécificités en phrases qui seront ensuite appliquées lors de la modélisation. Voilà les règles de métier de ce projet formule à l'aide de phrase.

RG1 : Un utilisateur peut importer plusieurs fichiers.

RG2 : Un fichier peut être importer par un seul utilisateur.

RG3 : Un utilisateur peut créer plusieurs discussions.

RG4 : Un discussions peut être créer par un seul utilisateur.

RG5 : Un utilisateur peut signaler plusieurs incidents.

RG6 : Un incidents est signaler par un seul utilisateur.

RG7 : Un incident peut avoir plusieurs log.

RG8 : Un log appartient à un incident.

RG9 : Un utilisateur peut envoyer plusieurs messages.

RG10 : Un message est envoyé par un seul utilisateur.

RG11 : Un utilisateur peut recevoir plusieurs messages.

RG12 : Un message peut être reçu par un seul utilisateur.

6.3 : Le diagramme de cas d'utilisation du système :

Pour dresser un diagramme de cas d'utilisation, il faut identifier les acteurs en premiers, puis les cas d'utilisation (actions) du système avec leur corrélation. La figure 6.1 représente le diagramme de cas d'utilisation globale du projet.



FIGURE 6.1: DIAGRAMME DE CAS D'UTILISATION DE GESTION DES TRAITEMENTS

6.4 : Les Diagrammes de séquences :

6.4.1 : Le Diagramme de séquence d'importation de fichier :

« Le diagramme d'importation de fichier » illustre les interactions entre les différents composants du système pour permettre l'importation réussie de fichier dans l'application.

TABEAU 6.2: DIAGRAMME DE SEQUENCE D'IMPORTATION DE FICHIER

Ligne	Description
Acteur concerné	Administrateur
Objectif	Enregistrement de toutes les données contenues dans un fichier
Pré condition	Avoir un compte existant
Scénario nominale	1 : Ouvrir formulaire d'importation de fichier 2 : Ouvrir explorateur de fichier 3 : Sélectionner un fichier 4 : Valider l'importation
Scénario d'exception	1 : Ouvrir formulaire d'importation de fichier 2 : Ouvrir explorateur de fichier 3 : Sélectionner un fichier 4 : Valider l'importation 5 : Affichage d'un message d'erreur

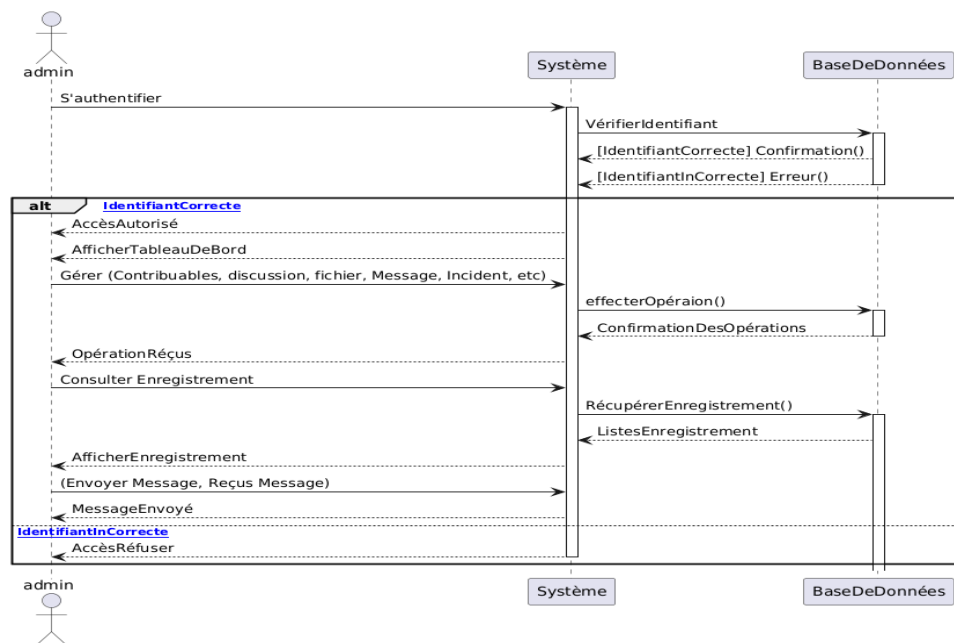


FIGURE 6.2: LE DIAGRAMME DE SEQUENCE GLOBALE DE L'ADMINISTRATEUR

6.4.2 : Le Diagramme de séquence d'authentification :

« Le diagramme de séquence d'authentification » décrit le processus par lequel un utilisateur et l'administrateur s'identifie auprès du système pour accéder au fonctionnalités de l'application.

TABLEAU 6.3: DIAGRAMME DE SEQUENCE D'AUTHENTIFICATION

Ligne	Description
Acteur concerné	Administrateur, enquêteur
Objectif	Accéder à des pages privilégiées selon les types d'utilisateurs
Pré condition	Avoir un compte existant
Scénario nominale	1 Ouvrir formulaire d'authentification et compléter champ 2 : Valider les informations saisies 3 : Accès aux pages demandées
Scénario d'exception	1 : Ouvrir formulaire d'authentification 2 : Saisir utilisateur, mot de passe 3 : Valider les informations saisies 4 : Information saisie erronée 5 : Affichage d'un message d'erreur

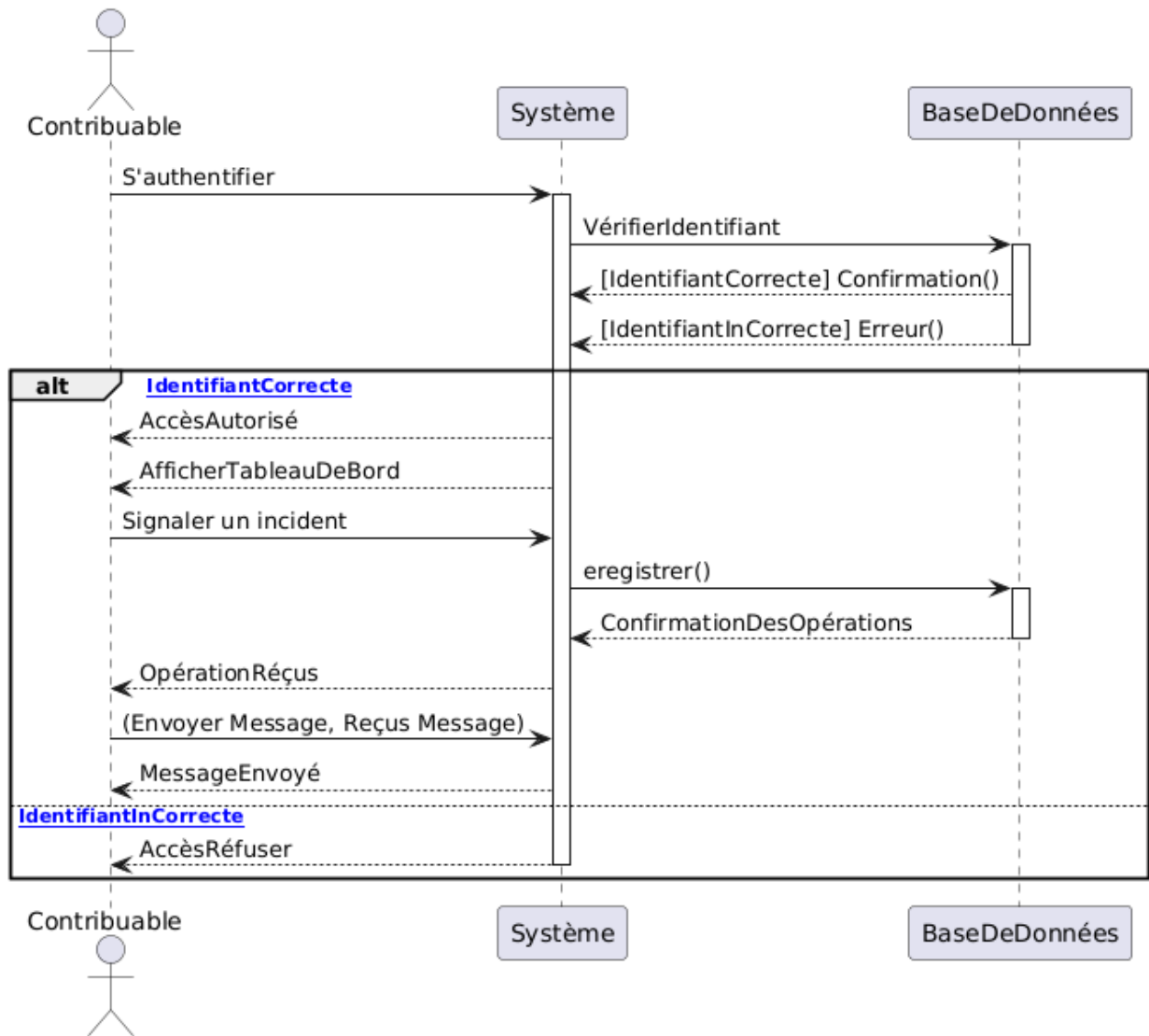


FIGURE 6.3: LE DIAGRAMME DE SEQUENCE GLOBAL DE L'UTILISATEUR

6.5 : Le Diagramme de classes :

« Le diagramme de classe » offre une vue statique des entités principales du système, leurs attribus, leurs méthodes, ainsi que les relations entre elles. Il constitue la base de la conception logicielle en identifiant les objets clés et leurs interactions.

La figure 6.4 représente le diagramme de classes de notre projet.

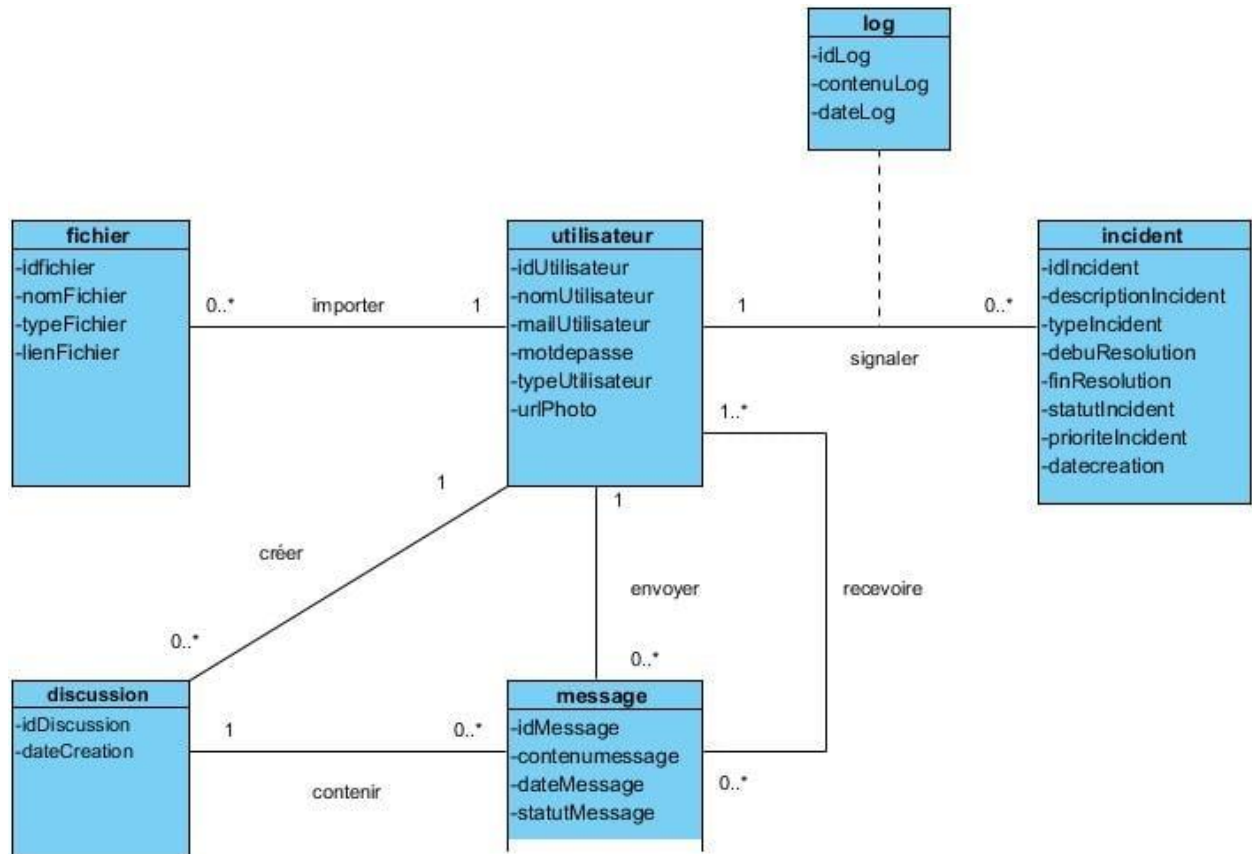


FIGURE 6.4: DIAGRAMME DE CLASSE DU PROJET

PARTIE III : REALISATION DU PROJET

Chapitre 7 : SPECIFICATION DES OUTILS DE REALISATION

Pour mettre en œuvre le site, il faut déterminer quelle technologie est la mieux adaptée en vue des existants et des buts à atteindre, il faut également choisir des outils adaptés à ces dernières afin de permettre une mise en œuvre réglementaire.

7.1 : Présentation des technologies utilisées

Puisqu'il est question ici de programmer un site web, il nous faut dans un premier temps choisir les outils et les technologies avec lesquels le développer, comme :

- Langage de programmation
- Frontend ou côté client
- Backend ou côté serveur
- Système de gestion de la base de données
- Serveur web
- Environnement de développement

7.2 : Technologie frontend

Le frontend est la partie visuelle de la partie visuelle de l'application web qui est affichée dans les navigateurs. L'architecture front-end repose sur les trois éléments principaux ci-dessous : -

- HTML
- CSS
- JavaScript

7.2.1 : HTML

L'HTML est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie « HyperText Markup Language », ce qui signifie en français « langage de balisage d'hypertexte ». Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme le texte et de placer des éléments interactifs, tel des liens, des images ou bien encore des animations. Ces éléments ne sont pas dans le code source d'une page codée en HTML mais "à côté" et la page en HTML ne fait que reprendre ces éléments.

Pour visualiser une page en HTML il est nécessaire d'utiliser un navigateur web.

7.2.2 : CSS

Le CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML. Introduit au milieu des années 1990, Il est aujourd'hui couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs.

Né en 1996, le CSS1 définit une cinquantaine de propriétés de rendu typographique du texte : polices, couleurs, tailles des caractères, les backgrounds image, marqueur de liste...

Rapidement complété par le CSS2, en 1997, ce langage se voit ajouter 70 propriétés supplémentaires, notamment des propriétés de positionnement. Ce complément sera malheureusement qualifié de précipité et son implémentation fortement ralentie, pour diverses raisons commerciales et de. C'est donc en 2007, sous appellation CSS 2.1, qu'il deviendra un standard que les navigateurs devront intégrer.

Fin 2011, le CSS3 est implémenté par les navigateurs et est perçu comme une très grande avancée. Contrairement à ses prédécesseurs, ce niveau est développé par modules, permettant ainsi aux navigateurs d'implémenter des sous-ensembles de CSS3.

Comme pour le langage HTML, les spécifications CSS ne sont pas développées par versions successives, mais par "niveaux", ce qui contraint chaque nouveau niveau à intégrer le précédent, et chaque implémentation à être compatible avec le niveau précédent.

CSS1 est donc développé pour être un sous-ensemble de CSS2, qui est lui-même développé pour être un sous-ensemble de CSS3.

7.2.3 : Javascript

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

Ainsi le langage Javascript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu.

Javascript a été mis au point par Netscape en 1995. A l'origine, il se nommait LiveScript et était destiné à fournir un langage de script simple au navigateur *Netscape*

Navigator 2. Il a à l'époque longtemps été critiqué pour son manque de sécurité, son développement peu poussé et l'absence de messages d'erreur explicites rendant dure son utilisation. Le 4 décembre 1995, suite à une association avec le constructeur Sun, Netscape rebaptise son langage Javascript (un clin d'oeil au langage Java développé par Sun). A la même époque, Microsoft mit au point le langage Jscript, un langage de script très similaire. Ainsi, pour éviter des dérives de part et d'autre, un standard a été défini pour normaliser les langages de script, il s'agit de l'ECMA 262, créé par l'organisation du même nom (ECMA, European Computer Manufacturers Association).

La figure 7.1 montre les navigateurs et le Javascript

Version de Javascript	Navigateurs
JavaScript 1.0	Netscape Navigator 2.0, Internet Explorer 3.0, Opera, Mozilla
JavaScript 1.1	Netscape Navigator 3.0, Opera, Mozilla
JavaScript 1.2	Netscape Navigator 4.0/4.05, Internet Explorer 4.0, Opera, Mozilla
Javascript 1.3	Netscape Navigator 4.06, Internet Explorer 5.0, Opera, Mozilla
JavaScript 1.4	Netscape Navigator 6.0, Internet Explorer 5.0, Opera, Mozilla
JavaScript 1.5	Netscape Navigator 6.0, Mozilla

FIGURE 7.1: NAVIGATEUR ET JAVASCRIPT

7.2.4 : Choix de l'outil

Javascript

Techniquement, ces trois éléments sont des lignes de code qui sont ensuite interprétées par le navigateur afin de produire les différentes images, couleurs, et autres éléments du site. Les activités de développement front-end sont assez fastidieuses et les développeurs peuvent passer de nombreuses et les développeurs peuvent passer de nombreuses heures à travailler pour arriver au résultat final.

Ainsi, il existe des assistants qui peuvent réduire la quantité de code requise et optimiser le processus de développement. Il en résulte une réduction du temps de développement et des coûts de mise sur le marché de produit. Ces assistants sont appelés des frameworks. Un framework comprend un code pré-écrit standardiser plus rapide. La quasi-totalité des frameworks modernes sont écrits en JavaScript. Parmi les plus populaires sont Angular et ReactJS.

Angular et React sont des ressources JavaScript puissantes et populaires pour construire à peu près tous les projets frontends que nous pouvons imaginer. Les deux facilitent

la construction d'interfaces utilisateur complexes et tendance pour les projets web. Par conséquent, ils ont souvent fait la une parmi les communautés de développeurs en ligne.

Il existe plusieurs technologies dans le monde de développeur qu'on peut utiliser pour notre projet.

On va voir une comparaison entre ReactJS et Angular, le plus célèbre. La comparaison entre ReactJS et Angular est représentée par le tableau 7.1

TABLEAU 7.1: COMPARAISON DE REACTJS ET ANGULAR

Paramètres	ReactJS	Angular
Groupe de soutien	Facebook	Google
Type	Librairie JavaScript	Framework complet
Reduction de code	Oui	Oui
Open source	Oui	Oui
Facilité d'apprentissage	Oui	Non
Simplicité	Oui	Non
Popularité	Plus populaire	Assez populaire
Langage	JavaScript	Typescript
Liberté	Limitée	Permet un choix de bibliothèques, d'architecture et d'outils

La figure 7.2 représente la popularité des technologies frontends.

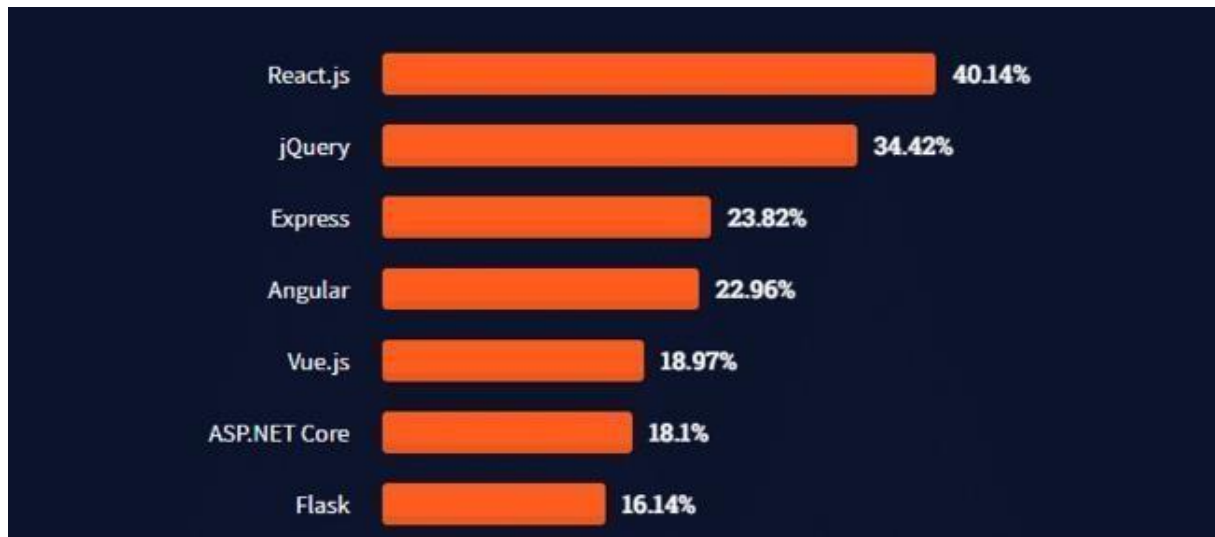


FIGURE 7.2: POPULARITE DES TECHNOLOGIES FRONTENDS

ReactJS est très populaire face au Angular et aux autres technologies frontend, Comme le tableau ci-dessus montre, ReactJS est le meilleur choix pour ce projet.

7.2.5 : Présentation de ReactJS

React est une Librairie JavaScript open source développée par une équipe de Facebook. React est couramment utilisé pour créer des interfaces utilisateur pour des applications à page unique à partir de composants isolés.

Il peut gérer les couches de vues des applications web et mobiles. Ainsi, React prend en charge le développement d'applications web et mobiles. Ce framework flexible permet aussi de créer des applications complexes si nous l'utilisons avec d'autres bibliothèques prises en charge.

React a une liaison de données unidirectionnelle, ce qui signifie que sa structure passe du parent à l'enfant. Cependant, pour une liaison de données bidirectionnelle, React propose `LinkStateMixin`, qui met en place le modèle général de boucle de flux de données.

Dans le flux de données traditionnel, pour chaque nouvelle entrée de données, il fallait recharger toute la page pour voir les modifications. Dans React, un rechargement n'est pas nécessaire. C'est parce que React ne crée pas de modèles d'objets de document (DOM) supplémentaires comme le ferait un flux de données traditionnel en acceptant de nouvelles mises à jour de données.

React doit être aussi associé à un framework pour construire des interfaces utilisateur rapides, belles et compatibles. C'est pourquoi, on a besoin d'un backend pour compléter le ReactJS

L'architecture d'un projet ReactJS est présentée par la figure 7.3

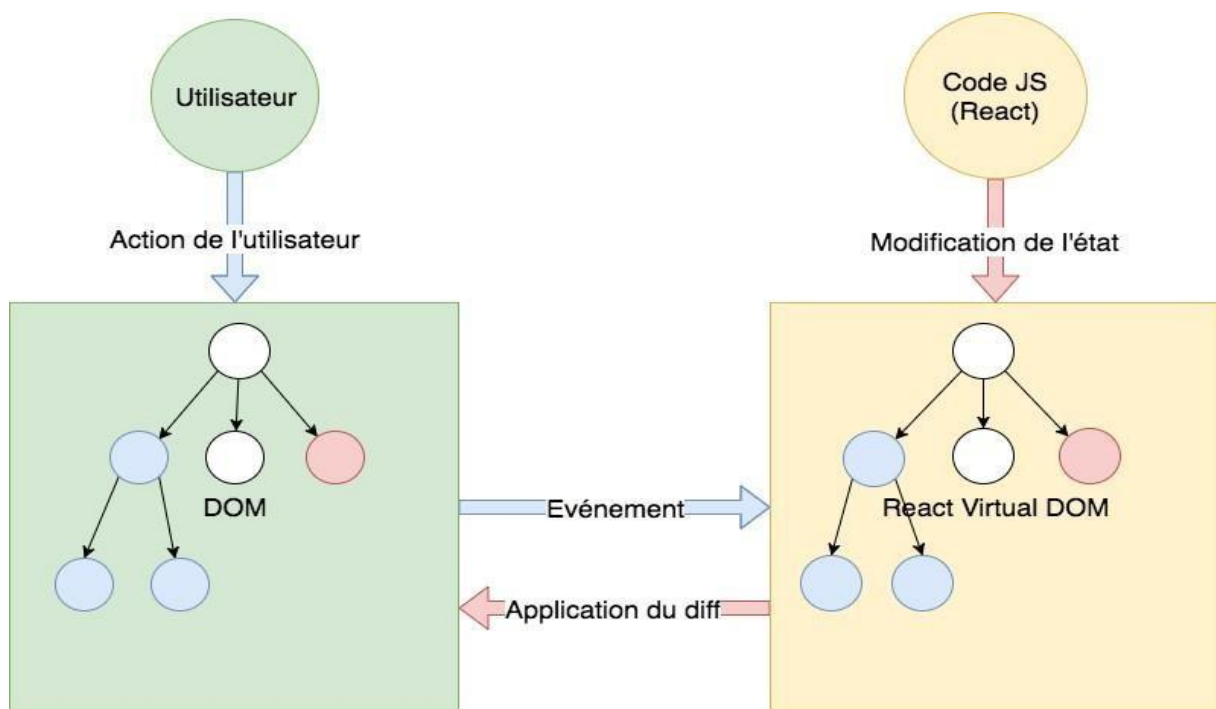


FIGURE 7.3: ARCHITECTURE D'UN PROJET REACTJS

7.3 : Technologie backend

Lorsque nous examinons les logiciels et les applications du point de vue d'un développeur, nous pouvons constater que de nombreuses choses se passent. Il y a une partie avant et une partie arrière.

La logique est présente, et tout ce qui concerne les opérations côté serveur et les fonctions de base de données est géré avec le backend. Bien que le frontend soit visible lorsque nous développons une application, la plupart des fonctionnalités de l'application se trouvent dans le backend.

Bien qu'il s'agisse de la partie centrale d'un logiciel ou d'une application, le backend n'est pas visible pour l'utilisateur final. La raison en est que le frontend est conçu pour rendre les choses moins complexes et plus faciles à utiliser.

L'utilisation de la meilleure technologie backend en termes de compatibilité soit une excellente chose, il n'est pas facile d'en choisir une. Il faut beaucoup de travail pour savoir quel langage est le meilleur pour une application logicielle. Il y a plusieurs langages de programmation côté serveur mais en parlant de facilité nous optons pour le langage PHP.

7.3.1 : Le langage PHP

Le PHP, pour Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit

d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.

Créé au début des années 1990 par le Canadien et Groenlandais Rasmus Lerdorf, le langage PHP est souvent associé au serveur de base de données MySQL et au serveur Apache. Avec le système d'exploitation Linux, il fait partie intégrante de la suite de logiciels libres LAMP.

7.3.2 : Framework

Framework est une sorte de cadre applicatif structurant qui permet de réduire le temps de développement des applications, tout en répondant de façon efficace aux problèmes rencontrés le plus souvent par les développeurs.

Il inclut généralement de nombreuses fonctionnalités prêtes à l'emploi dont les implémentations sont bien rodées et utilisent des modèles de conception standard et réputés. Le temps ainsi gagné sur les questions génériques pourra être mis à profit pour les parties spécifiques de l'application.

Enfin un framework c'est aussi le fruit du travail de dizaines de personnes qui s'appliquent à corriger les problèmes ou les failles de sécurité découvertes par l'ensemble des utilisateurs et à proposer de nouvelles fonctionnalités. De ce fait, les programmes d'un Framework sont en général mieux conçus et mieux codés, mais aussi mieux débogués et donc plus robustes que ce que pourrait produire un unique programmeur. Outre le gain de temps, on obtient un important gain en termes de qualité.

7.3.3 : Choix du framework backend

Parmi les framework backend on a Symfony , Laravel, CodeIgniter, Yii,.. Symfony et Laravel sont les deux frameworks PHP populaires, donc c'est à partir de ces deux framework que nous allons faire notre choix. Ces frameworks PHP rendent le processus de développement plus facile, plus rapide et plus flexible que le Core PHP. En outre, ils permettent aux développeurs d'adapter facilement un système. Ils améliorent également la réutilisation et la maintenabilité du code et rendent l'application plus sécurisée. La comparaison entre Symfony et Laravel est illustrée par le tableau 7.2

TABEAU 7.2: COMPARAISON DE SYMFONY ET LARAVEL

Paramètre	Symfony	Laravel
Communauté	Grande	Grande
MVC	Oui	Oui
Langage	PHP	PHP
Sécurité	Oui	Oui
Fiabilité	Oui	Oui
Moteur de gabarit	Template Twig	Gabarit Blade
Accès à la base de données	Utilisation de doctrine pour accéder à la base de données.	Eloquent pour accéder à la base de données.
Popularité	Grande	Grande
Migration	Automatique	manuelle

La différence entre Symfony et Laravel est que Symfony est un framework d'applications Web PHP avec un ensemble de bibliothèques et de composants PHP réutilisables, tandis que Laravel est un framework Web PHP libre et open source basé sur Symfony. En conclusion, les deux cadres rendent le processus de développement plus rapide et plus facile.

Comme nous n'avons que trois (03) mois pour mettre en place ce site, donc Symfony est le meilleur choix car sa migration de données est automatique.

7.3.3.1 : Présentation de Symfony

Grâce à Symfony, la rapidité de développement est accrue du fait de ses nombreux composants permettant de ne pas réinventer la roue et d'automatiser beaucoup de tâches standards (authentification, vérification des données, couche sécurité de l'application). Le développement d'applications est aussi beaucoup plus maintenable grâce à la stabilité apportée par la couche Symfony, qui a aussi l'avantage d'éviter beaucoup d'optimisations manuelles grâce à sa gestion du cache. Les projets sont plus structurés, ce qui réduit énormément le temps alloué à la maintenance ou à des évolutions. Le transfert des connaissances est par conséquent beaucoup plus simple.

Il a été créé en 2005 par SensioLabs, une agence Web française, et plus précisément par Fabien Potencier. Il l'a développé à l'origine pour ses propres besoins car elle devait toujours redévelopper les mêmes fonctionnalités.

7.3.3.2 : Structure MVC

Symfony est un framework MVC (Modèle-vue-contrôleur qui est une architecture logicielle que nous détaillerons).

L'architecture MVC est l'une des plus utilisées pour les applications Web. Elle consiste à découper l'application en modules qui ont trois responsabilités différentes :

- Les modèles gèrent l'accès aux données (le plus souvent dans une base de données).

Les vues sont les interfaces graphiques présentées à l'utilisateur. Les contrôleurs contiennent la logique des actions à effectuer suivant la requête de l'utilisateur.

La figure 7.4 illustre la structure MVC.

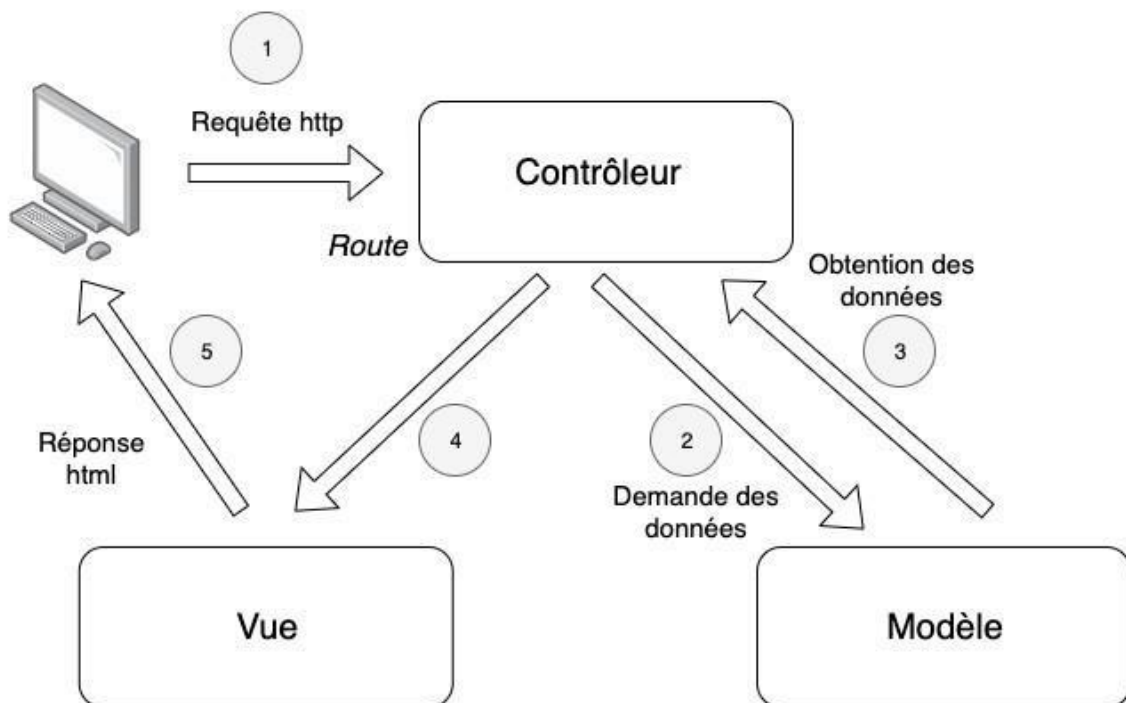


FIGURE 7.4: STRUCTURE MVC

L'utilisateur envoie une requête HTTP, par exemples POST /inscription.

Le routeur de l'application va appeler le bon contrôleur en fonction de la route (c'est-à-dire en fonction du verbe utilisé et de l'URI). Dans notre exemple, ce sera le contrôleur responsable de l'inscription d'un utilisateur.

Le contrôleur est responsable de coordonner les actions à effectuer. Ici par exemple, il peut vérifier le contenu de la requête, puis demander au modèle d'inscrire l'utilisateur dans la base

de données. Une fois que le modèle aura inscrit l'utilisateur, il pourra préparer la vue à retourner à l'utilisateur.

Le modèle est responsable de la validation de la lecture et de l'enregistrement des données. Dans notre exemple, il va s'assurer que toutes les données pour créer un utilisateur ont bien été fournies. Dans l'affirmative il va créer l'utilisateur dans la base de données et le passer au contrôleur.

La vue est responsable de l'interface graphique. Elle contient la logique pour l'affichage des données qui ont été récupérées dans par le modèle.

7.4 : Combinaison de ReactJS avec Symfony

Cet écosystème Symfony offre aussi de nombreuses garanties, comme la structuration du code qui "normalise" les projets et permet leur reprise par d'autres développeurs, une bonne maintenabilité, et une architecture très extensible.

Pour la partie visible des projets, c'est-à-dire le "front-end", nous utilisons de plus en plus le framework open source ReactJS (porté par Facebook). En effet, depuis quelques années, de nombreuses librairies javascript commencent à s'imposer dans les projets pour offrir de la réactivité, des interfaces fluides, une maintenabilité facilitée, et une totale autonomie avec les composants serveurs, notamment Symfony. Les projets sont plus structurés, ce qui réduit énormément le temps alloué à la maintenance ou à des évolutions. Le transfert des connaissances est par conséquent beaucoup plus simple. Les projets Symfony sont bien structurés et très agréables à manipuler. Malgré la rigueur nécessaire pour se servir de ce framework, il reste très flexible et plaisant à utiliser.

Symfony possède un module qui permet d'intégrer la librairie ReactJS. Ce composant s'appelle

« webpack Encore ». Pour la récupération des données, nous avons utilisé un API de ReactJS « axios »

7.4.1 : Webpack Encore

Webpack est un module bundle open source. Son objectif principal est de regrouper des fichiers JavaScript pour les utiliser dans un navigateur. Cet outil est également capable de transformer, regrouper ou emballer à peu près n'importe quelle ressource.

Le but de Webpack est de parcourir les fichiers, de trouver les relations entre les fichiers et de fournir des solutions pour packager tous les fichiers qu'il trouve ensemble. Lorsqu'il identifie des fichiers JS, il sait les manipuler, les concaténer. S'il rencontre des

éléments qui ne sont pas du JavaScript, alors il a besoin de loader pour savoir les interpréter. De base, Webpack ne sait que manipuler du JS mais il inclut aussi des plugins de minification. Ainsi quand on lance Webpack en mode production, il va automatiquement faire des optimisations avec de la minification.

7.4.2 : Axios

Lors de la création d'une application Web, il est fréquent que nous souhaitons utiliser et afficher les données provenant d'une API. La librairie Axios est une librairie JavaScript permettant d'interagir avec une API. Nous pouvons donc effectuer toutes les opérations de base, et cela comprend la récupération des données. Axios utilise les promesses, une fonctionnalité spécifique du langage JavaScript. Il est donc bien nécessaire de comprendre comment elles fonctionnent pour manipuler la librairie.

Les promesses représentent le résultat d'une fonction asynchrone. Avec la librairie Axios, tous les appels sont ainsi effectués de manière asynchrone. Pour manipuler les données récupérées, nous devons donc attendre que l'appel soit terminé. Il est possible d'exécuter du code après le résultat d'une promesse grâce à la méthode "then". Cette méthode accepte en paramètre une fonction. Nous pouvons donc transmettre à cette fonction la réponse de votre API. L'attribut "data" va contenir les données récupérées. ReactJS s'occupe d'afficher ces données une fois ils sont reçu.

7.5 : Système de gestion de base de données SGBD

Tous les données statistiques doivent être stocké dans une base de données pour avant d'être analysé ou traiter.

Un système de gestion de base de données(SGBD) est le logiciel qui permet à un ordinateur de stocker, récupérer, ajouter, supprimer et modifier des données. Un SGBD gère tous les aspects primaires d'une base de données, y compris la gestion de la manipulation des données, comme l'authentification des utilisateurs, ainsi que l'insertion ou l'extraction des données. Un SGBD définit ce qu'on appelle le schéma de données ou la structure dans laquelle les données sont stockées.

Dans cette optique, les SGBD sont utilisés pour interagir avec ces enregistrements et réaliser des opérations CRUD (Creat, Read, Uptad, Delete), et de recherche.

7.5.1 : . Le choix du SGBD

Puisqu'il est question d'utiliser une base de données pour stocker les données liées au site, il est important de choisir quel SGBD utiliser, chaque SGBD dispose de ses atouts et de

ses avantages, le tout est d'analyser chaque SGBD en tenant compte des contraintes qui découlent du site.

Pour cela, avant de passer en revue les différents SGBD à disposition, il est important d'évaluer la capacité requise par le projet mis en œuvre.

Considérons deux SGBD les plus utilisés, tel que : MySQL, PostgreSQL, ...

A partir du tableau 7.3 nous prouvons que le MySQL est le meilleur choix pour ce projet

TABEAU 7.3: COMPARAISON DE MYSQL ET POSTGRESQL

Caractéristique	MySQL	PostgreSQL
Gratuit	OUI	OUI
Rapidité	OUI	OUI
Multiplateforme	OUI	OUI
Facilité d'utilisation	OUI	NON
Sécurité	OUI	OUI
Fonctionnalité	MOYENNE	MOYENNE
Facilité de l'installation	OUI	NON

Comparé aux autres, pour ces caractéristiques, l'utilisation et l'installation de MySQL est facile, il est sécurisé comme PostgreSQL.

7.5.2 : MySQL

MySQL est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle. Une base de données n'est qu'une collection structurée de données qui est organisée pour en faciliter l'utilisation et la récupération.

MySQL a été lancé à l'origine en 1995. Depuis, il a connu quelques changements de propriétaire et de gestion, avant de se retrouver chez Oracle Corporation en 2010. Alors qu'Oracle est en charge maintenant, MySQL est toujours un logiciel open source.

7.5.3 : XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (**X** (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand

nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

7.5.3.1 : Serveur web

Les serveurs de fichiers, les serveurs de bases de données, les serveurs de messagerie et les serveurs web utilisent différents types de logiciels de serveur. Chacune de ces applications peut accéder aux fichiers stockés sur un serveur physique et les utiliser pour différents buts.

Le travail d'un serveur web consiste à servir des sites web sur internet. Pour atteindre cet objectif, il agit comme un intermédiaire entre le serveur et les machines des clients. Il extrait le contenu du serveur sur chaque requête d'utilisateur et le transmet au web.

Le plus grand défi d'un serveur web est de servir simultanément plusieurs et différents utilisateurs web – chacun demandant des pages différentes. Les serveurs web traitent les fichiers écrits le langage de programmation tel que PHP.

Ils les transforment en fichiers HTML statiques et diffusent ces fichiers dans le navigateur des utilisateurs web. Quand nous entendons le mot serveur web, considérez-le comme l'outil responsable de la communication serveur-client.

Comme serveur web, nous avons choisi Apache car celui-ci est intégré dans XAMPP qui est le Serveur d'application choisie.

7.5.3.1.1 : *Apache*

Le logiciel libre Apache HTTP Server (Apache) est un serveur HTTP créé et maintenu au sein de la fondation Apache. Jusqu'en avril 2019 ce fut le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Bien que nous appelions Apache un serveur web, ce n'est pas un serveur physique mais plutôt un logiciel qui s'exécute sur un serveur. Son travail consiste à établir une connexion entre un serveur et les navigateurs des visiteurs du site web (Firefox, Google Chrome, Safari, etc.) tout en délivrant des fichiers entre eux (structure client-serveur). Apache est un logiciel multiplateforme, il fonctionne donc à la fois sur les serveurs Unix et Windows.

7.5.4 : Outil de conception

Pour concevoir ce projet nous avons besoins d'un outil de conception. Il existe plusieurs outils de conception. Pour concevoir un système d'information, nous avons adopté un outil de conception tel que : Visual Paradigm, Win'Design, Paradigm Plus, Windev, Poseidon for UML, Visual UML. Nous allons présenter deux outils de conception.

7.5.4.1 : Visual Paradigm

Visual paradigm permet la création des diagrammes UML et des modèles qui en sont à l'origine. Ceux-ci peuvent alors générer du code dans un langage de programmation déterminé. Il propose également la création d'autres types de diagrammes, comme celui qui permet la modélisation des bases de données pouvant, lui aussi, générer des canevas d'applications basés sur des Framework et Pattern mais en plus, générer du code SQL qu'il peut ensuite déployer automatiquement dans différents environnements. Visual paradigm assure aussi la gestion des exigences, l'analyse d'impacts, qui permet de connaître à l'avance les conséquences d'un changement et la création de rapports automatisés. Il dispose des outils de paramétrage permettant de définir et de mettre en forme les types d'objets utilisés dans les modélisations.

7.5.4.2 : StarUML

StarUML est un outil de génie logiciel dédié à la modélisation UML et édité la société coréenne MKLabs. Il est multiplateforme et fonctionne sous Windows, Linux et MacOS. .StarUML offre des fonctions de modélisation orientée objet basée, avec un support complet d'UML, conforme aux normes UML 2.x.

7.5.5 : Choix d'outil de conception

Nous allons dresser un tableau 7.4 pour comparer les outils de conception pour mieux faire le choix entre eux.

TABEAU 7.4: COMPARAISON DE VISUAL PARADIGME ET STARUML

Caractéristique	Visual Paradigm	StarUML
Fiabilité	OUI	OUI
Efficacité	OUI	OUI
Validité	OUI	OUI
Portabilité	OUI	OUI
Enregistrement sans licence	NON	OUI

A partir de ce tableau, nous avons décidé d'utiliser StarUML qui est le meilleur outil pour la conception de notre application.

7.5.6 : Environnement de développement

Un environnement de développement intégré, ou IDE, est un logiciel de création d'applications, qui rassemble des outils de développement fréquemment utilisés dans une seule interface utilisateur graphique (GUI). Un IDE se compose habituellement des éléments suivants :

- Éditeur de code source : un éditeur de texte qui aide à la rédaction du code logiciel, avec des fonctions telles que la coloration syntaxique avec repères visuels, la saisie automatique en fonction du langage et la vérification de bogues dans le code pendant la rédaction.
- Utilitaires d'automatisation de version locale : des utilitaires qui permettent d'automatiser des tâches simples et reproductibles lors de la création d'une version locale du logiciel à destination du développeur lui-même, par exemple la compilation du code source en code binaire, la mise en paquet du code binaire et l'exécution de tests automatisés.
 - Débogueur : un programme qui permet de tester d'autres programmes en affichant l'emplacement des bogues dans le code d'origine.

7.5.6.1 : Visual studio code

Comme nous allons développer une application basée sur Symfony et ReactJS, Visual studio. Code est le meilleur choix.

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés constituent un freeware, c'est-à-dire un logiciel gratuit pour toute utilisation mais privé.

Chapitre 8 : MISE EN ŒUVRE ET IMPLEMENTATION

La présentation de l'application est le chapitre final afin de montrer le résultat du projet, d'après les besoins, les choix d'outils de réalisation et de développement et les Framework utilisées pour l'achever.

8.1 : Architecture logicielle

L'architecture logicielle décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions.

Pour définir :

- Elle se consacre à structurer et à concevoir une application à partir de ses spécifications fonctionnelles.
- Elle structure et décompose de façon logique chaque application en couches.
- Elle introduit les notions et concepts de découpage en couches, modules, composants, design patterns et Framework.

La figure 8.1 illustre l'architecture logicielle du projet.

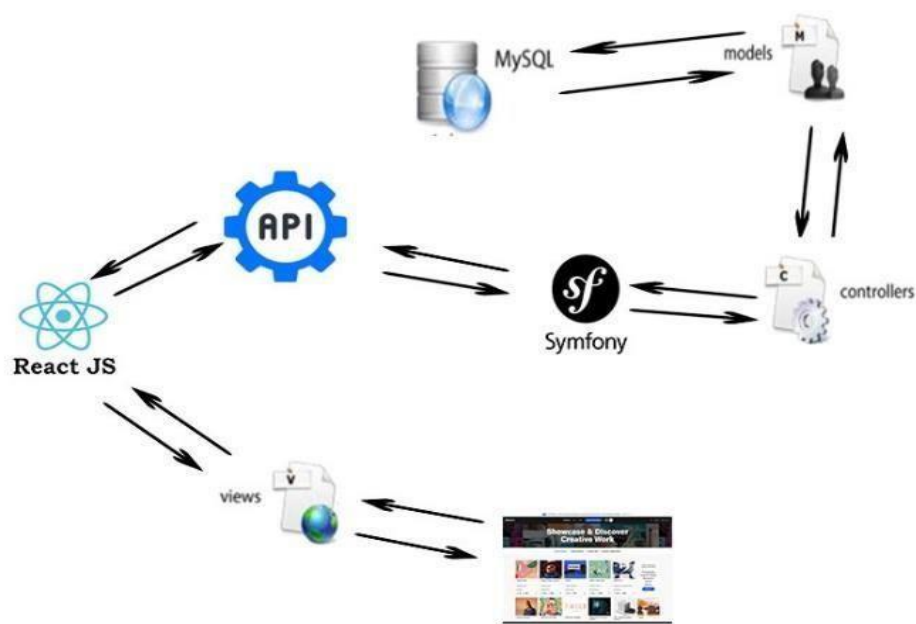


FIGURE 8.1: ARCHITECTURE LOGICIELLE

8.2 : Architecture matérielle

En informatique, architecture désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et /ou matériels et /ou humains et/ou informations) et des relations entre les éléments. Notre architecture ne s'agit ici que d'une architecture matérielle de base requise, et peut parfaitement être sujet à d'éventuelles extensions. L'architecture matérielle est représentée par la figure 8.2

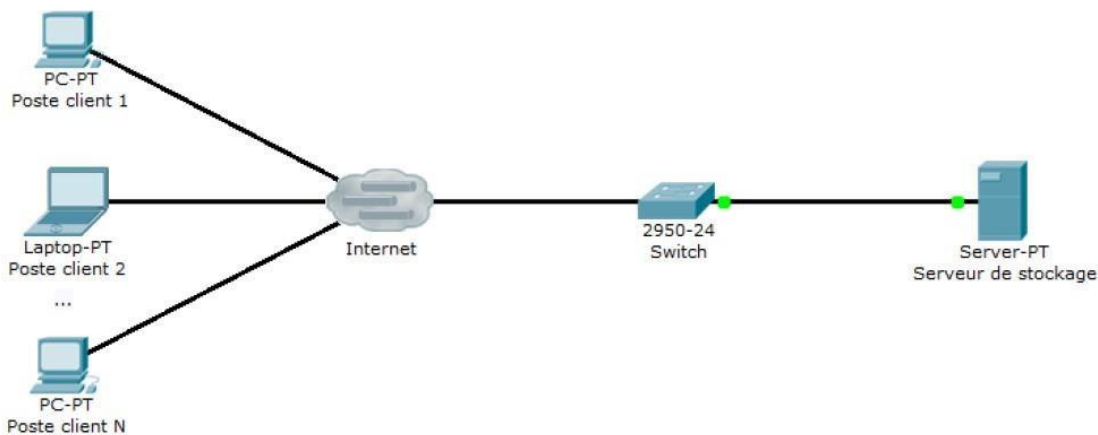


FIGURE 8.2: ARCHITECTURE MATERIELLE

8.3 : Présentation des différentes parties de l'application web

Il existe deux type d'utilisateur dont : enquêteur et administrateur.

8.3.1 : Enquêteur

Les enquêteurs ne peuvent pas accéder à toutes les fonctionnalités de l'application. Ils peuvent seulement accéder au page contenant le formulaire de l'insertion.

8.3.2 : Administrateur

L'utilisateur administrateur accède à toutes les fonctionnalités de l'application.

8.4 : Conception de l'IHM

Il y a trois participants dans l'interface homme-machine, à savoir :

- L'utilisateur (participant avec choix),
- La machine (participant avec programme),
- Le concepteur (participant qui anticipe les choix possibles de l'utilisateur et les code dans un programme).

8.4.1 : Création de menus et interfaces

La partie visuelle de l'application est l'intermédiaire entre l'utilisateur et la base de données. C'est la partie immédiatement visible pour l'utilisateur et elle mérite une attention. Il doit être facile à comprendre et facile à utiliser.

L'entête et le pied de page et le contenu de la page comme celle illustrée. L'entête de page sera constitué d'une bannière avec le logo.

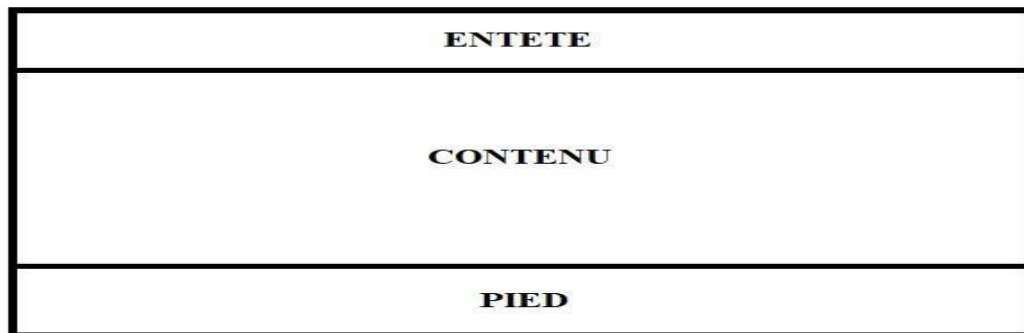


FIGURE 8.3: MAQUETTE IHM DE L'APPLICATION

8.4.2 : . Fonctionnalité

La fonctionnalité de base de toute IHM (Interface Homme Machine) est de permettre une interaction entre les utilisateurs de l'application et celui-ci. La population visée par cette application est large et son niveau présumé en informatique est basique. Il est donc primordial de concevoir une interface :

- Agréable esthétiquement ;
- De prise en main rapide ;
- Intuitive ;
- A l'ergonomie classique, c'est-à-dire non déroulante.

De même, il est absolument inenvisageable de considérer une IHM qui ne permettrait pas le contrôle total de l'application : données, paramètres, etc. Pour une utilisation simple et effective depuis tout appareil, nous avons choisi d'utiliser le principe de la responsivité. Donc cette application web peut s'adapter sur n'importe quel appareil que ce soit sur Desktop ou mobile.

Mais la seule contrainte pour l'utilisateur étant de disposer d'un navigateur web.

Chapitre 9 : PRESENTATION DE L'APPLICATION DEVELOPPEE

Après établi et effectue la différente conception, et enfin choisir l'outil et le langage à utiliser, on peut entamer la programmation. Dans cette étape, on procède à la présentation de l'application qui permettra de montrer les difficultés rencontrées qui nous ont permis de dégager un ensemble de pistes susceptibles de nous aider.

9.1 : Page d'authentification :

Lors du lancement de l'application, une page de connexion apparait. L'utilisateur et l'Administrateur doivent s'authentifier en saisissant son nom et son mot de passe. Si les données saisies sont correctes, alors la page de l'application qui contient le menu s'affiche. Si l'une de données saisies est incorrecte, l'application ne peut pas s'ouvrir et rester sur la page d'authentification.

La figure 9.1 nous présente la page d'authentification.

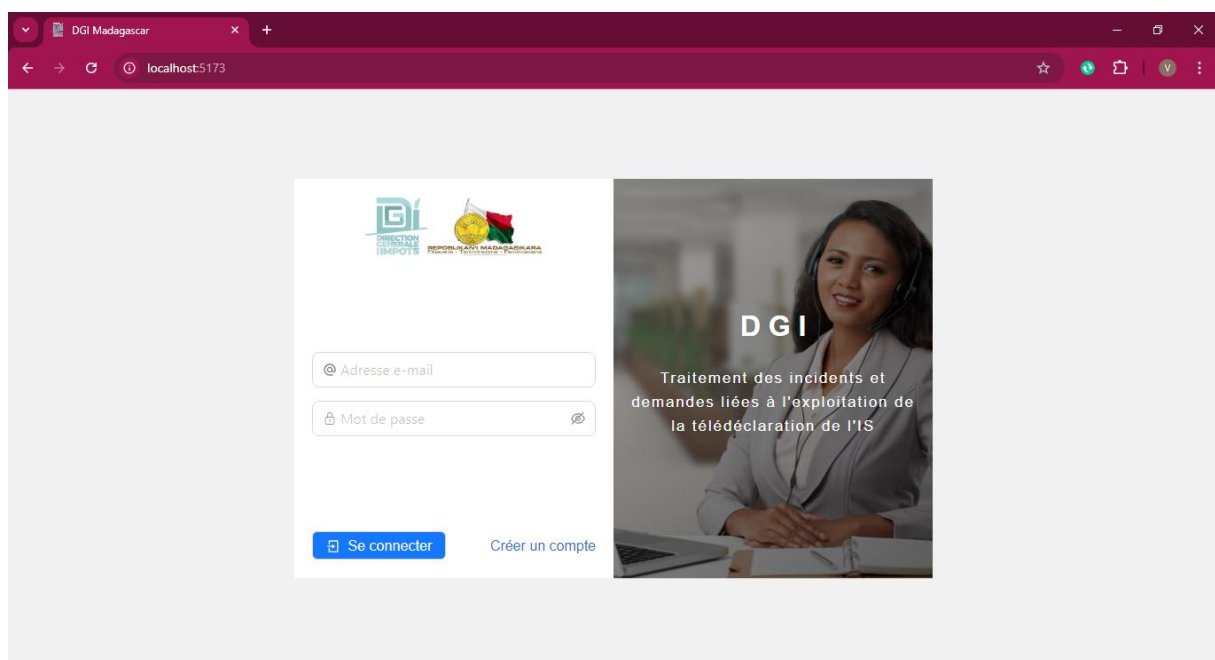


FIGURE 9.1: PAGE D'AUTHENTIFICATION

9.2 : COTE ADMINISTRATEUR

9.2.1 : Page du tableau de bord :

C'est ici dans cette page qu'on va voir le diagramme des incidents.

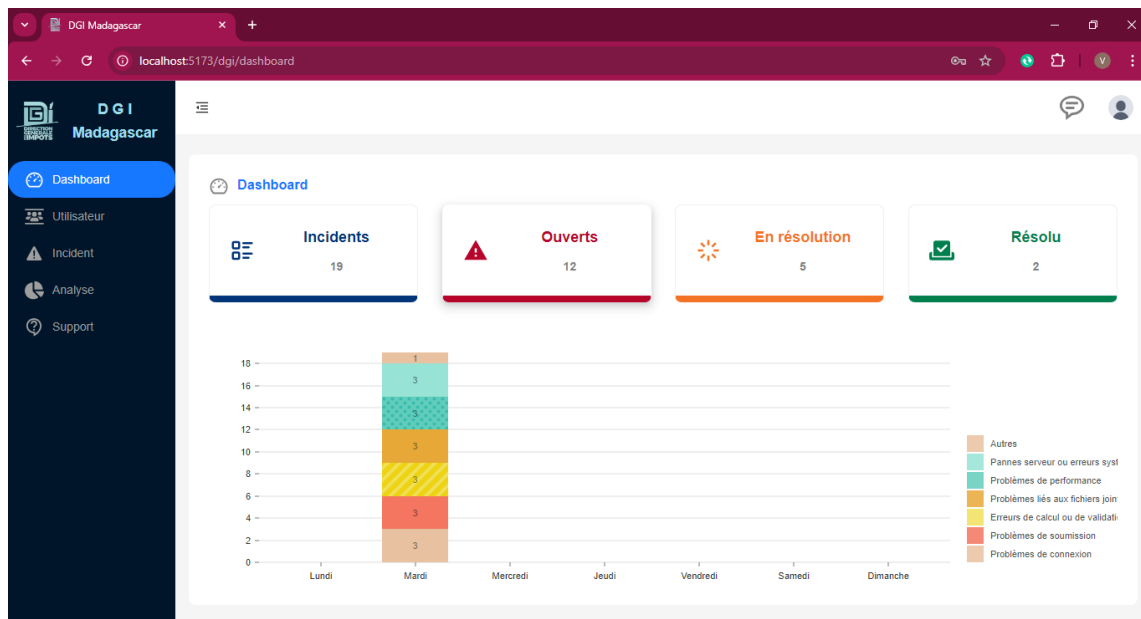


FIGURE 9.2: PAGE DU TABLEAU DE BORD

9.2.2 : Liste de l'Utilisateur :

Dans cette liste, l'administrateur seul peut regarder la liste des utilisateurs. Donc cette liste c'est la liste des utilisateurs.

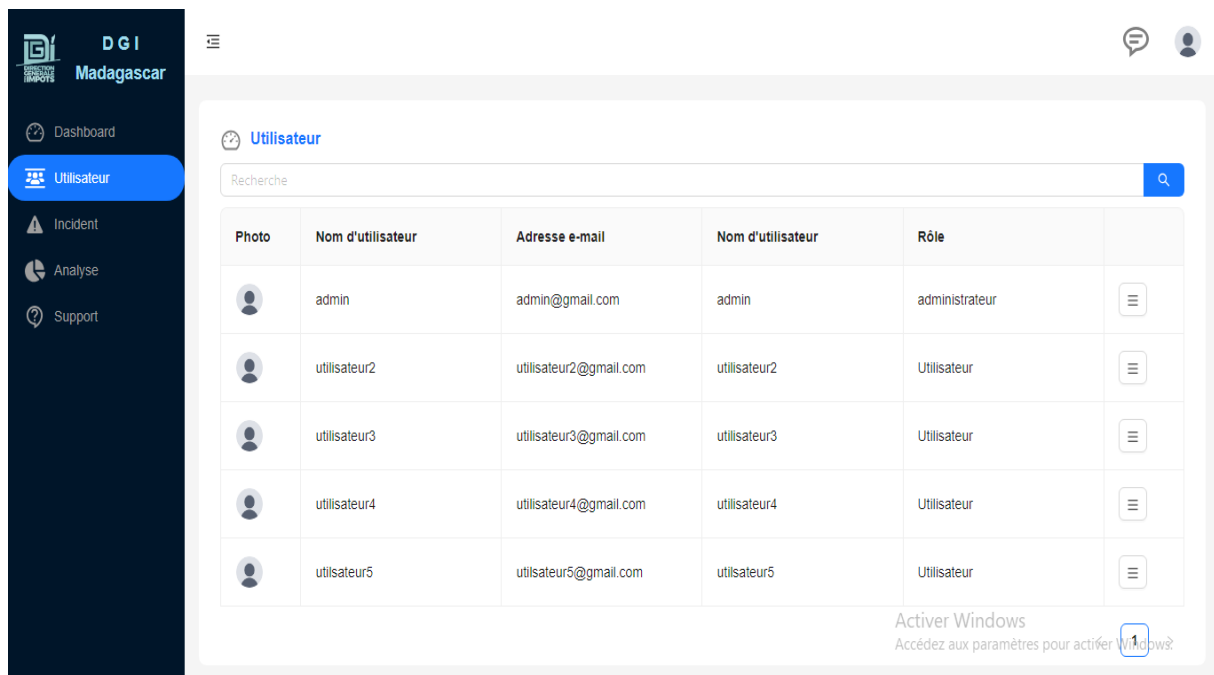


FIGURE 9.3: LISTE DES UTILISATEURS

9.2.3 : Liste des Incidents :

Ce sont les listes de tous les incidents signalés de la part des utilisateurs qu'on va voir dans cette page.

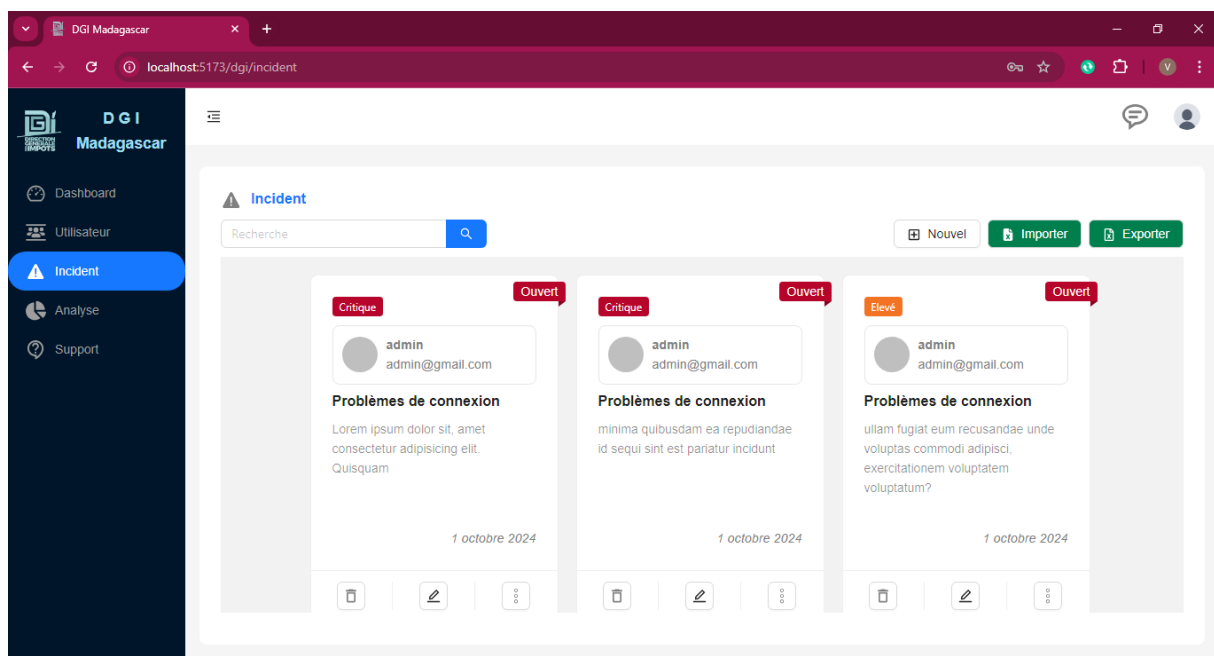


FIGURE 9.4: LISTE DES INCIDENTS

9.2.4 : Diagramme de l'Analyse :

On voit dans le diagramme de l'analyse quand il y a des utilisateurs ont signalés des incidents et ces incidents sont affichés par semaines.

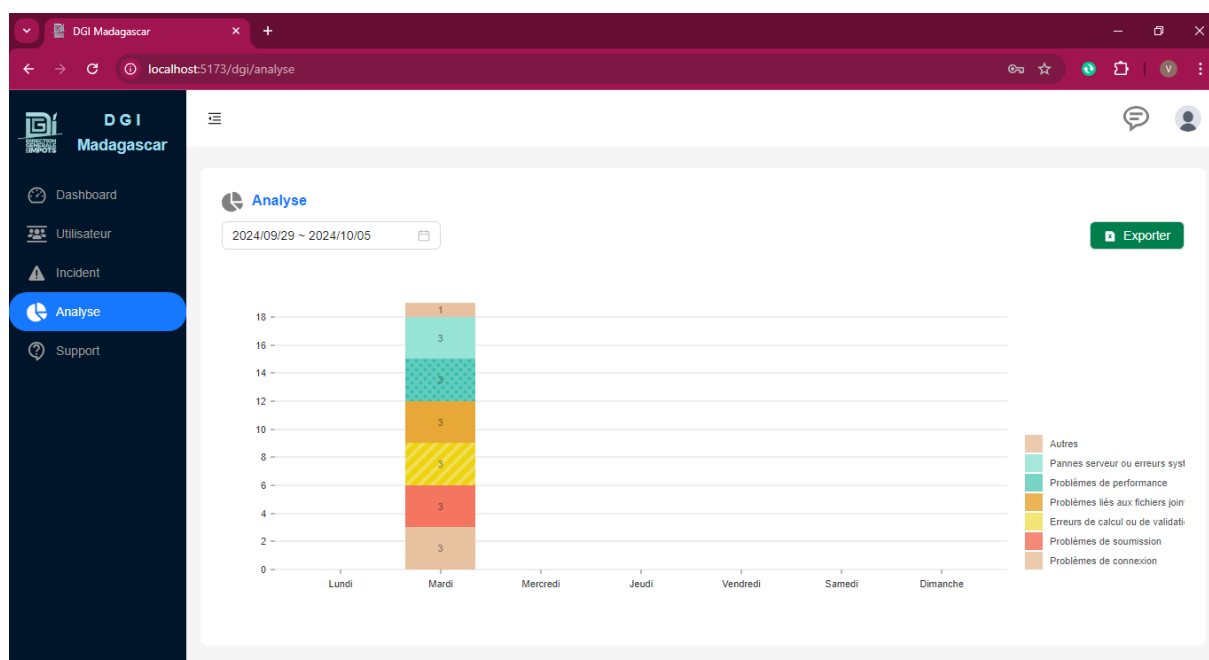


FIGURE 9.5: DIAGRAMME DE L'ANALYSE

9.2.5 : Liste des supports :

C'est ici que l'administrateur et les utilisateurs peuvent se parler entre eux. L'admin ne peut pas crée la discussion car le principe c'est quand l'utilisateur à des problèmes, il y a le seul peut crée de discussion et l'administrateur à juste répondre à son message et résoudre ses problèmes.

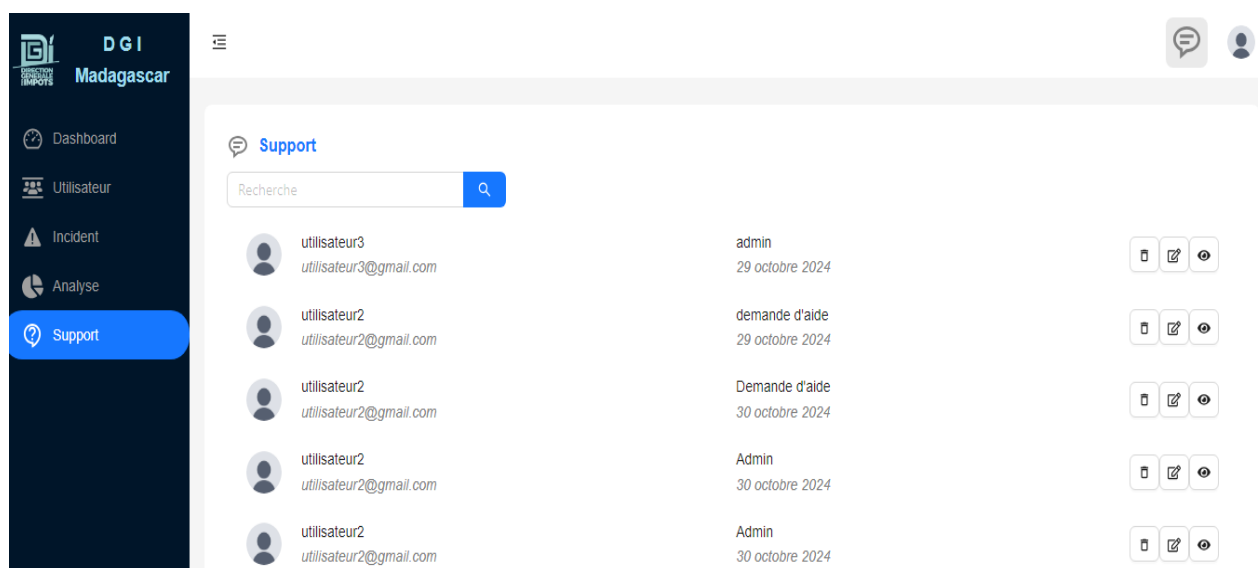


FIGURE 9.6: LISTE DES SUPPORTS

9.3 : COTE UTILISATEUR

9.3.1 : Liste des incidents du coté utilisateur :

Dans cette page qu'on va voir les types des incidents qui ont signalés par chaque utilisateurs.

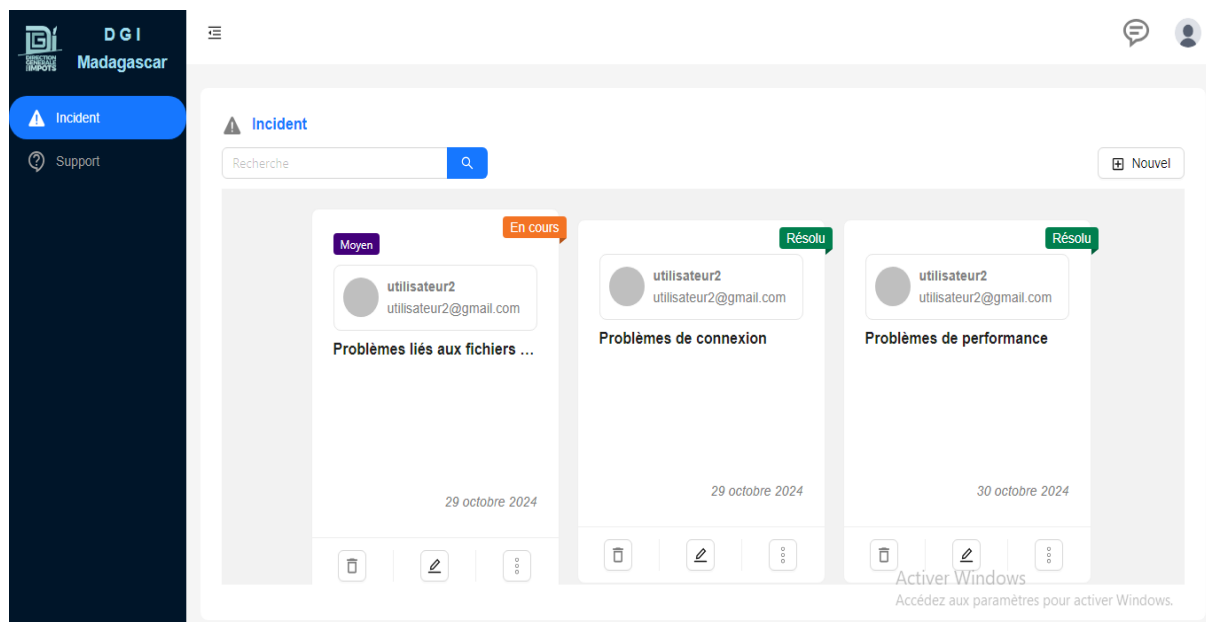


FIGURE 9.7: LISTE DES INCIDENTS DU COTE UTILISATEURS

9.3.2 : Listes de support du coté utilisateur :

C'est ici que l'utilisateur peut créer une discussion avec l'administrateur s'il y a des problèmes.

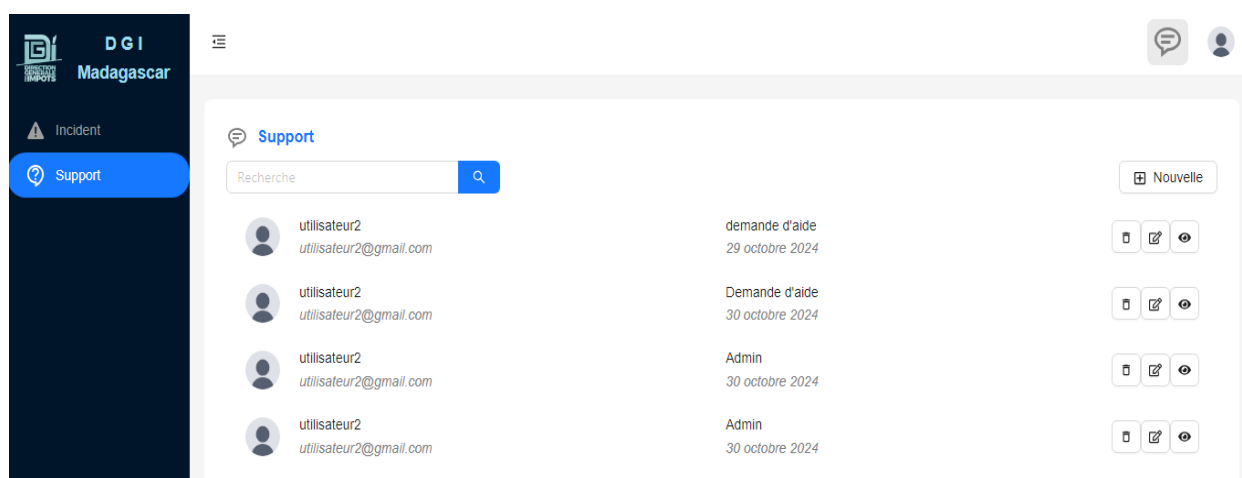


FIGURE 9.8: LISTE DES SUPPORT DU COTE UTILISATEUR

Ensuite, la page suivante montre leurs discussions.



FIGURE 9.9: PAGE DE DISCUSSION

CONCLUSION GENERALE ET PERSPECTIVE

Pour conclure, l'évolution de la technologie informatique engendre un grand changement dans la vie d'une entreprise. Cette offre vise les avantages de l'Impôt synthétique (IS) dans la DRI sur le Traitement des incidents et demandes liée à l'exploitation du télé déclaration de l'IS.

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, l'étude de l'existant jusqu'à la réalisation de ce logiciel, j'ai adopté le notation « UML » et de maîtriser le nodeJs et Framework reactJs que nous avons reçus au sein de l'EMIT dont nous en sommes fier.

Malgré cela, l'application que nous avons créés n'a pas atteint sa performance parfaite. Des améliorations doivent être envisagées afin d'augmenter cette exploit. Les agents de la DRI doivent avoir des rôles et des responsabilités clairement définis pour gérer efficacement ces incidents et demandes. Il est également important d'évaluer et d'améliorer continuellement les processus de traitement afin d'assurer une qualité élevée du service aux contribuables.

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

- ❖ [Bakari M., 2023], *Support de cours conduite de projet informatique*, Troisième année en Développement d'Application Internet Intranet.
- ❖ [Jacques Aimé., 2023], *Support de cours de JavaScript*, Troisième année en Développement d'Application Internet Intranet.
- ❖ [Crampes J.-B., 1997]. *Interfaces graphiques ergonomiques – Conception et Modélisation*, Ellipses, collection TECHNOSUP / Génie Logiciel ; Document *fiche de poste de MNP*,
- ❖ Étude comparative des différents cycles de vie [en ligne]. Disponible sur : < <http://www.over-blog.com/article-etude-comparative-des-différents-cycles-de-vie-delogicielle> >
- ❖ [ARISETRA, 2023], *cours Génie Logiciel*, Troisième année en Développement d'Application Internet Intranet.
- ❖ Introduction à ASP[en ligne]. The Best Cours. Disponible sur : < <http://www.bestcours.com/318-introduction-asp.net-2010> >
- ❖ [Jean-François P., 2022,] *Ergonomie d'une application web* [en ligne]. Comment Ça Marche. Disponible sur : < <http://www.commentcamarche.net/contents/1255-ergonomie-d-une-application-web> >
- ❖ [Jean-François P., 2020], *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> >
- ❖ [Jean-Michel D., 2012,] *Développons en Java Les JSP (Java Server Pages)* [en ligne]. Développons en Java. Disponible sur : < <http://www.jmdoudoux.fr/java/dej/chap-jsp.htm> >
- ❖ [Hasina, 2023] *Cours UML (Unified Modeling Language)*. Troisième année en Développement d'Application Internet Intranet.
- ❖ Cours UML sur Comment ça marche.net
- ❖ [Lai M., 1998.] *Penser objet avec UML*, Inter Editions ;
- ❖ [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;

- ❖ [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> >
- ❖ [Muller P.-A., 1998.] *Modélisation objet avec UML*, Eyrolles ;
- ❖ [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.
- ❖ [Romain B., 2016.] Démarrez avec wampserver [en ligne]. WampServer. Disponible sur < <http://www.wampserver.com/#begin-wrapper> >
- ❖ [Rumbaugh J., Jacobson I. et Booch G., 1999]. *The Unified Modeling Language Reference Manual*, Addison-Wesley.
- ❖ [Hasina ,2022] *Cycle de vie du logiciel et bonnes pratiques de développement*.

ANNEXE

INSTALLATION DU FRAMEWORK REACT.JS

Avant d'installer React, assurez-vous qu'ordinateur. Une fois que vous l'avez installé, exécutez la commande ci-dessous dans votre terminal :

« **node -v** »

Si tout s'est bien passé, la commande ci-dessus devrait vous indiquer quelle version de Node.js est actuellement installée sur votre ordinateur. Ensuite, nous allons exécuter une autre commande pour commencer à installer React. Tout d'abord, créez un dossier ou naviguez jusqu'à l'emplacement où vous voulez que l'application React soit installée, puis exécutez la commande ci-dessous dans votre terminal :

« **npx create-react-app react-kit** »

La commande ci-dessus va installer React dans un dossier appelé **react-kit**. Une fois l'installation terminée, ouvrez votre dossier React nouvellement installé dans l'éditeur de code de votre choix. Nous utiliserons Visual Studio Code dans ce tutoriel. Visual Studio Code est livré avec un terminal intégré. Si vous comptez utiliser un autre terminal comme Git Bash ou CMD, il vous suffit de naviguer dans le répertoire de votre application React et d'exécuter la commande ci-dessous :

« **npm run start** »

Cette commande met en route votre serveur de développement. Après un petit moment, vous devriez avoir cette page ci-dessous rendue sur **localhost :3000** dans votre navigateur :



SERVEUR DE DEVELOPPEMENT REACT APRES INSTALLATION

Si vous avez suivi jusqu'à ce point, alors vous avez installé avec succès votre première application React. Félicitations !

Lorsque vous jetez un coup d'œil à votre structure de dossiers, vous verrez un dossier appelé **public**. Ce dossier contient un fichier **index.html** où notre code sera injecté et servi au navigateur. Le dossier **src** est l'endroit où se trouveront toute notre logique, notre style et notre balisage ; c'est notre dossier de développement avec le fichier **App.js** faisant office de composant racine. Le code de ce fichier est ce que nous avons dans l'image ci-dessus. Ouvrez le fichier **App.js** et apportez quelques modifications au fichier, puis enregistrez pour voir qu'il se reflète automatiquement dans le navigateur.

Les composants dans React

Il existe deux principaux types de composants dans React : les composants de classe et les composants fonctionnels. La documentation de React est actuellement en cours de réécriture à l'aide de Hooks, qui est une fonctionnalité que l'on trouve dans les composants fonctionnels. Nous utiliserons également les composants fonctionnels dans ce tutoriel car ils sont devenus les composants les plus utilisés pour les applications React.

Dans la plupart des cas, les composants renvoient du code HTML avec un mélange de valeurs dynamiques créées à l'aide de JavaScript. Les composants sont créés comme des fonctions en JavaScript.

Comment installer Node.js

Si vous avez déjà installé Node.js, vous pouvez passer cette section.

SUR MACOS

En utilisant Bash sur Mac

```
Curl "https://nodejs.org/dist/latest/node-${VERSION:-$(wget -qO-  
https://nodejs.org/dist/latest/ | sed -nE 's|.*>node-(.*)\.pkg.*|1|p')}.pkg" >  
"$HOME/Downloads/node-latest.pkg" && sudo installer -store -pkg  
"$HOME/Downloads/node-latest.pkg" -target "/"
```

En utilisant Homebrew sur Mac

Brew install node

Sur Windows

Téléchargez le programme d'installation Windows directement depuis le site Web nodejs.org.

Sur Linux

Sur Ubuntu 18.04+, vous pouvez installer Node en utilisant les commandes suivantes.
`sudo apt update`
`sudo apt install nodejs`

Vérifier la version de Node.js installée

Dans la section précédente, nous avons installé Node. Vérifions-le en contrôlant la version installée. Exécutez la commande suivante dans le terminal.

« **node -v** »

Vous pouvez également le vérifier via la commande plus longue

« **node --version** »

Vous devriez voir une sortie similaire à celle-ci. Bien que le numéro de version puisse varier. «
V14.16.0 »

RESUME

Le thème traite du traitement des incidents et des demandes liées à l'exploitation du télé déclaration de l'IS dans la DRI de Fianarantsoa. Il est important de définir clairement les rôles et responsabilités des agents pour gérer efficacement ces incidents et demandes. Il faut également évaluer et améliorer continuellement les processus de traitement afin d'assurer une qualité élevée du service aux contribuables.

Enfin, la formation et la sensibilisation des agents à la gestion des incidents et des demandes liées à l'exploitation du télé déclaration de l'IS sont des éléments clés pour réduire le risque de problème potentiels et garantir la pérennité du système.

MOTS CLES : Application Web, Télé-déclaration, Impôt Synthétique (IS), ReactJs, UML.

ABSTRACT

The theme deals with the treatment of incidents and requests related to the operation of the IS télédéclaration in the DRI of Fianarantsoa. It is important to clearly define the roles and responsibilities of agents to effectively manage these incidents and requests. It is also necessary to continuously evaluate and improve the processing procedure to ensure high-quality service to taxpayers.

Finally, the training and sensitization of agents in incident management and requests related to the operation of the IS télédéclaration are key elements to reduce the risk of potential problems and ensure the sustainability of the system.