# CURRICULUM VITAE

# CURRICULUM VITAE



telephone 034 51 264 08

boite-de-reception [rakotoalisonmirantsoaniaina@gmail.com](mailto:rakotoalisonmirantsoaniaina@gmail.com)

home013E/3603 Antarandolo Fianarantsoa

**COMPÉTENCES :**

**COMPÉTENCES :**

**Linguistiques** :

* Français : Bien,
* Anglais : Assez-bien.

**En informatique** :

* ***Langage de programmation*** :
* Frontend : HTML/CS3, React ;
* Backend: JAVA, NODE.js/express.js, MySQL;
* ***Méthode de CONCEPTION :***

MERISE

Notation UML

**CENTRES D'INTÉRÊTS :**

**CENTRES D'INTÉRÊTS :**

* **Chanter,**
* **Voyager,**
* **Natation,**

**RAKOTOALISON Mirantsoa Niaina**

Etudiante en troisième année de licence professionnelle à l’Ecole Nationale d’Informatique (ENI), et suis le parcours Génie Logiciel de base de données.

**FORMATIONS ET DIPLÔME OBTENUS :**

**FORMATIONS ET DIPLÔME OBTENUS :**

**FORMATIONS ET DIPLÔME OBTENUS :**

**FORMATIONS ET DIPLÔME OBTENUS :**

**2021-2022** : Troisième année de Licence professionnelle en génie Logiciel et base de données à l’Ecole Nationale d’Informatique Fianarantsoa ;

**2020-2021** : Deuxième année de Licence professionnelle en génie Logiciel et base de données à l’Ecole Nationale d’Informatique Fianarantsoa ;

**2018** : Obtention du diplôme de Baccalauréat série D mention Bien.

**EXPÉRIENCES PROFESSIONNELLES :**

Figure : Organigramme de la direction générale des douanes malagasy**EXPÉRIENCES PROFESSIONNELLES :**

Figure : Organigramme de la direction générale des douanes malagasyFigure : Organigramme de la direction générale des douanes malagasy**EXPÉRIENCES PROFESSIONNELLES :**

Figure : Organigramme de la direction générale des douanes malagasy**EXPÉRIENCES PROFESSIONNELLES :**

Figure : Organigramme de la direction générale des douanes malagasy**EXPÉRIENCES PROFESSIONNELLES :**

Figure : Organigramme de la direction générale des douanes malagasyFigure : Organigramme de la direction générale des douanes malagasy**EXPÉRIENCES PROFESSIONNELLES :**

**Projet FRONTEND Mars et Mai 2022 - SAYNA :**

* Création de trois sites : MARVEL, BATMAN, EVENT
* Technologies : HTML5/CSS3/BOOTSTRAP/JAVASCRIPT

**Projet GESTION de prestation Médecin - ENI** :

* Développement logiciel
* Technologies : JAVAFX, MySQL
* Outils : WampServer, NetBeans

**Projet GESTION de la paie du SERVICE Financier** :

* Développement logiciel
* Technologies :VB, SQL
* Outils : Microsoft Access

# SOMMAIRE GENERAL

[CURRICULUM VITAE I](file:///C:\Users\rakot\Downloads\UNIVERSITE%20DE%20FIANARANTSO1.docx#_Toc125110629)

[SOMMAIRE GENERAL II](#_Toc125110630)

[REMERCIEMENTS III](#_Toc125110631)

[LISTE DES FIGURES IV](#_Toc125110632)

[LISTE DES TABLEAUX VI](#_Toc125110633)

[LISTE DES ABREVIATIONS VII](#_Toc125110634)

[INTRODUCTION GENERALE 1](#_Toc125110635)

[PARTIE I : PRESENTATION 2](#_Toc125110636)

[Chapitre 1 : PRESENTATION DE L’ECOLE NATIONALE D’INFORMATIQUE 2](#_Toc125110637)

[1.1. Information d’ordre générale 2](#_Toc125110638)

[1.2. Missions et historiques 2](#_Toc125110639)

[1.3. Organigramme institutionnel de l’ENI 4](#_Toc125110640)

[1.4. Domaines et spécialisation 7](#_Toc125110641)

[1.5. Architecture des formations pédagogiques 7](#_Toc125110642)

[1.6. Relation de l’ENI avec les entreprises et organismes 10](#_Toc125110643)

[1.7. Partenariat au niveau international 11](#_Toc125110644)

[1.8. Débouchés professionnels des diplômés 13](#_Toc125110645)

[1.9. Ressources humaines 15](#_Toc125110646)

[Chapitre 2 : Présentation de l’établissement d’accueil 17](#_Toc125110647)

[1.1. Information d’ordre générale 17](#_Toc125110648)

[1.2. Historique 17](#_Toc125110649)

[1.3. Mission 22](#_Toc125110650)

[1.4. Organigramme 22](#_Toc125110651)

[Chapitre 3 : Description du projet 23](#_Toc125110652)

[3.1. Formulation 23](#_Toc125110656)

[3.2. Objectif et besoins de l’utilisateur 23](#_Toc125110657)

[3.3. Moyens nécessaires à la réalisation du projet 24](#_Toc125110663)

[3.4. Résultats attendus 25](#_Toc125110664)

[PARTIE II : ANALYSE ET CONCEPTION 2](#_Toc125110665)

[Chapitre 4 : Analyse préalable 24](#_Toc125110666)

[4.1. Analyse de l’existant 24](#_Toc125110671)

[4.2. Critique de l’existant 25](#_Toc125110672)

[4.3. Conception avant-projet 26](#_Toc125110673)

[Chapitre 5 : Analyse et Conception 36](#_Toc125110674)

[5.1. Dictionnaire des données 36](#_Toc125110680)

[5.2. Règles de gestion 39](#_Toc125110681)

[5.3. Représentation et spécification des besoins 42](#_Toc125110689)

[5.4. Modélisation du domaine 51](#_Toc125110690)

[Chapitre 6 : Conception détaillée 62](#_Toc125110691)

[6.1. Architecture du système 62](#_Toc125110698)

[6.2. Diagramme de séquence de conception pour chaque cas d’utilisation 63](#_Toc125110699)

[6.3. Diagramme de classe de conception pour chaque cas d’utilisation 75](#_Toc125110700)

[6.4. Diagramme de classe de conception global 80](#_Toc125110701)

[6.5. Diagramme de paquetages 82](#_Toc125110702)

[6.6. Diagramme de déploiement 82](#_Toc125110703)

[PARTIE III : REALISATION 83](#_Toc125110704)

[Chapitre 7 : Mise en place de l’environnement de développement 83](#_Toc125110705)

[7.1. Installation et configuration des outils 83](#_Toc125110708)

[7.2. Architecture de l’application 86](#_Toc125110709)

[Chapitre 8 : Développement de l’application 87](#_Toc125110710)

[8.1. Création de la base de données 87](#_Toc125110711)

[8.2. Codage de l’application 89](#_Toc125110712)

[8.3. Présentation de l’application 92](#_Toc125110713)

[CONCLUSION 96](#_Toc125110714)

[BIBLIOGRAPHIE VIII](#_Toc125110715)

[WEBOGRAPHIE IX](#_Toc125110716)

[GLOSSAIRE X](#_Toc125110717)

[RESUME XV](#_Toc125110718)

[ABSTRACT XV](#_Toc125110719)

# REMERCIEMENTS

En premier lieu, je remercie Dieu tout puissant, de m’avoir donné la santé et l’aptitude, ainsi que l’audace pour dépasser toutes les difficultés durant ce stage. Je tiens aussi à remercier, respectueusement,

* Monsieur HAJALALAINA Aimé Richard, Docteur HDR et Président de l’Université de Fianarantsoa, de m’avoir considéré parmi les étudiants de l’Université de Fianarantsoa.
* Monsieur MAHATODY Thomas, Docteur HDR et Directeur de l’École Nationale d’Informatique Fianarantsoa, de m’avoir permis de réaliser ce stage ;
* Monsieur LAINKANA Zafivanona Ernest, Directeur Général de la Douane, de m’avoir accepté comme stagiaire chez la Douane ;
* Madame RAHANTASOA Yollande, Directeur de la STATISTIQUE ET DE LA COMPTABILITE de la Douane, de m’avoir accepté comme stagiaire dans sa direction ;
* Monsieur RANDRIANARIVELO Hasina, chef de service de l’informatique, de m’accorder comme parmi les stagiaires dans son service ;
* Monsieur RANDRIAMIAKATRA Andry Hubert, chef de division autres applications auprès du service de l’informatique, de m’avoir accepté comme stagiaire et aussi de m’avoir encadré professionnellement ;
* Monsieur, RAKOTOASIMBAHOAKA Cyprien Robert, Maître de Conférence, d’être à la Présidence du Jury ;
* Monsieur, RAZAFIMAHATRATRA Hajarisena, Docteur en informatique, d’être mon examinateur ;
* Monsieur RALAIVAO Jean Christian, Assistant d’Enseignement Supérieur, d’être mon encadreur pédagogique ;
* Tout le personnel de la douane qui m’a aidé durant mon investigation.
* Ma famille qui m’a aidée moralement et financièrement.
* Mes proches qui m’ont aidé et soutenu durant ces périodes de stage et de préparation de cette mémoire.

# LISTE DES FIGURES

# LISTE DES TABLEAUX

[Tableau 1: Organisation du système de formation pédagogique de l’Ecole 6](#_Toc125094508)

[Tableau 2:Liste des formations existantes à l’ENI 8](#_Toc125094509)

[Tableau 3:Débouchés professionnels éventuels des diplômés 13](#_Toc125094510)

[Tableau 4: Chronogramme des activités 23](#_Toc125094511)

[Tableau 5: Moyens matériels 25](#_Toc125094512)

[Tableau 6 : Comparaison entre les solutions 27](#_Toc125094513)

[Tableau 7: Comparaison entre MERISE et Unified process 28](#_Toc125094514)

[Tableau 8: Comparaison de STARUML et VISUAL PARADIGM 31](#_Toc125094515)

[Tableau 9: comparaison entre deux SGBD existants 32](#_Toc125094516)

[Tableau 10 : Comparaison entre PHP et JAVA 34](#_Toc125094517)

[Tableau 11 : Comparaison les deux outils utiles pour le développement Web 35](#_Toc125094518)

[Tableau 12 : Le dictionnaire des Données 36](#_Toc125094519)

[Tableau 13: Identification des acteurs 44](#_Toc125094520)

[Tableau 14: Identification des messages 45](#_Toc125094521)

[Tableau 15: Description des cas d'utilisation du système 47](#_Toc125094522)

[Tableau 16: Priorisation des cas d'utilisation du système 49](#_Toc125094523)

# LISTE DES ABREVIATIONS

**2TUP** : 2 Track Unified Process

**ASR :** Administration des Systèmes et Réseaux

**CNH :** Commission Nationale d’Habilitation

**CUR :** Centre Universitaire Régional

**DSC :** Direction des Statistiques et de la Comptabilité

**ENI :** Ecole National d’informatique

**GB :** Génie logiciel et base de données

**HDR :** Habilitation à Diriger des Recherches

**IDE :** Integrated Environnement Developpement

**IG :** Informatique Général

**IRD :** Institut de Recherche pour le Developement

**MVC :** Model View Controller

**PRESUP :** Programme de Renforcement de l’Enseignement Supérieur

**SDC :** Service de la Comptabilité

**SGBD :** Système de Gestion de Base de données

**SI :** Service de l’Informatique

**SSAE :** Service des Statistiques et des Analyses Economiques

**UML:** Unified Modeling Language

**UP:** Unified Process

# INTRODUCTION GENERALE

La technologie ne cesse d’impressionner le monde grâce au très haut niveau d’optimisation qu’elle apporte, mais aussi la mobilité et la communication qu’elle favorise. Même les entreprises étant acteurs majeurs de gagner confiance, d’attirer l’attention, de fidéliser ses clients pour apporter un grand essor économique l’utilise pour atteindre ces objectifs.

Cas de la douane Malagasy, une administration bicentenaire ayant un sacré nombre d’anciennetés, possédant un slogan très riche : « Douane, une administration innovante, redevable, partenaire pour l’émergence du pays », y fait confiance et l’adopte pour réaliser ses objectifs. L’existence d’un service de l’informatique peut être une preuve directe pour le prouver, en addition, la présence d’une division de développement d’autres applications qui se charge de prévoir un projet utile pour l’optimisation des activités de chaque employé de la douane.

Ici cette division a pu prévisualiser, que les employés de la douane peuvent parfois affronter des problèmes qui bloquent et mènent au non achèvement de leurs travaux. Par conséquent, elle décide de mettre en place un genre de système de ticketing au niveau de la douane pour remédier à cet obstacle.

Et c’est précisément le rôle qui m’a été confié, d’assurer la réalisation et conception d’une application web utile aux interventions à chaque problème affronté par les employés de la douane. Elle leurs permet : de signaler des problèmes liés à leurs travaux, de repérer facilement les agents à qui s’adresser dans un tel ou tel problème, de visualiser des solutions si un problème de même cas est déjà résolu. Elle permet aussi aux intervenants de faire des interventions plus rapides et faciles sans avoir répéter la même intervention à chaque circonstance identique précédemment.

Donc, pour sa conception, une méthode sera adoptée ; et pour sa réalisation, on utilisera un système de gestion de base de données, et aussi des langages de programmation appropriés au développement web.

Ainsi, en premier lieu, on va présenter l’établissement d’origine du stagiaire, l’ENI, et l’entreprise où le stage a été effectué, la Douane Malagasy ; en second lieu, on va faire part des phases d’analyse et de conception ; puis, on achèvera le développement de ce rapport par la description des étapes de la réalisation du projet avant de conclure.

# PARTIE I : PRESENTATION

## Chapitre 1 : PRESENTATION DE L’ECOLE NATIONALE D’INFORMATIQUE

Ce chapitre porte la présentation de l’Ecole Nationale d’Informatique détaillant ses informations, missions, domaine et spécialisation, les débouchés des diplômés ainsi que ses ressources humaines.

### Information d’ordre générale

L’Ecole Nationale d’Informatique, en abrégé ENI, est un établissement d’enseignement supérieur rattaché académiquement et administrativement à l’Université de Fianarantsoa. Le siège de l’Ecole se trouve à Tanambao- Antaninarenina à Fianarantsoa.

Les coordonnées pour la prise de contact avec l’Ecole sont les suivantes :

Ecole Nationale d’Informatique (ENI)

Antaninarenina Tanambao

BP 1487 FIANARANTSOA (301)

Téléphones : 034 05 733 36 ou 033 42 302 02

Adresse mail : [eni@eni.mg](mailto:eni@eni.mg)

Site : [www.eni.mg](http://www.eni.mg)

### Missions et historiques

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd’hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L’Ecole s’est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l’ENI était constituée et créée au sein du (CUR) par le décret N° 83-185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d’Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L’ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

En fournissant à des étudiants des connaissances de base en informatique ;

En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.

En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l’information et de la communication (TIC).

L’implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l’a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l’emploi.

La filière de formation d’Analystes Programmeurs a été mise en place à l’Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d’ingénieurs a été ouverte à l’Ecole en 1986.

Dans le cadre du Programme de renforcement en l’Enseignement Supérieur (PRESUP) la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l’appui matériel et financier de la Mission Française de coopération auprès de l’Ambassade de France à Madagascar.

Une formation pour l’obtention de la certification CCNA et / ou NETWORK + appelée « CISCO Networking Academy » a été créée à l’Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l’Ecole Supérieure Polytechnique d’Antananarivo (ESPA). Cependant, cette formation n’avait pas duré longtemps.

L’organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d’embauche pour les diplômés de l’Ecole:

* Génie Logiciel et base de Données.
* Administration des Système et réseaux.

La mise en place à l’Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l’effectif des étudiants accueillis à l’Ecole, notamment à cause du manque d’infrastructures, un système de « Formation Hybride » a été mise en place à partir de l’année 2010. Il s’agit en effet d’un système de formation semi-présentielle et à distance avec l’utilisation de la visioconférence pour la formation à distance.

Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu’Université de Toliara.

### Organigramme institutionnel de l’ENI

Cet organigramme de l’Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983.

Sur cet organigramme, l’Ecole placée sous la tutelle académique et administrative de l’Université de Fianarantsoa, est dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l’Etablissement et nommé par un décret pris en Conseil des Ministres pour un mandat de 3 ans.

Le Conseil de l’Ecole est l’organe délibérant de l’Ecole.

Le Collège des Enseignants regroupant tous les enseignants-chercheurs de l’Ecole est chargé de résoudre les problèmes liés à l’organisation pédagogique des enseignements. Il propose et coordonne les programmes d’activités pédagogiques.

Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l’établissement, en tenant compte notamment de l’évolution du marché de travail et de l’adéquation des formations dispensées par rapport aux besoins des entreprises. Il coordonne les programmes de recherche à mettre en œuvre à l’Ecole.

Le Secrétariat principal coordonne les activités des services administratifs (Scolarité, Comptabilité, et Intendance).

Conformément aux textes en vigueur régissant les Etablissements malgaches d’Enseignement Supérieur, qui sont barrés sur le système LMD, les Départements de Formation pédagogique ont été ainsi remplacés par des Mentions et des parcours. Et les chefs des Départements ont été ainsi remplacés par des responsables des mentions et les responsables des parcours.

L’organigramme de l’Ecole Nationale d’Informatique est présenté par la Figure 8

Figure 1 : Organigramme de l'ENI

**CONSEIL D’ECOLE**

**DIRECTION**

**CONSEIL SCIENTIFIQUE**

**COLLEGE DES ENSEIGNANTS**

**SERVICE PEDAGOGIQUE**

RESPONSABLE DE MENTION

CHEFS DE PARCOURS :

GB – SR - IG

**SECRETARIAT**

**SCOLARITE**

CHEF DE SERVICE SCOLARITE

**INTENDANCE**

DEPOSITAIRE COMPTABLE

RESPONSABLE ENTRETIEN

**SERVICE ADMINISTRATIF ET FINANCIER**

SECRETARE PRINCIPAL

**SECRETARIAT DE DIRECTION**

SECRETAIRE DE DIRECTION

COMPTABLE

**SECRETARIAT**

**SECURITE ET AFFAIRES ENVIRONNEMENTALES**

GARCONS DE SALLES / FEMMES DE MENAGE / SECURITE

### Domaines et spécialisation

Les activités de formation et de recherche organisées à l’ENI portent sur les domaines suivants :

* Génie logiciel et Base de Données ;
* Administration des Systèmes et Réseaux ;
* Informatique Générale ;
* Modélisation informatique et mathématique des Systèmes complexes.

D’une manière plus générale, les programmes des formations sont basés sur l’informatique de gestion et sur l’informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d’Informatique fondamentale que des éléments d’Informatique appliquée

Le tableau 1 décrit l’organisation du système de formation pédagogique de l’Ecole.

Tableau :Organisation du système pédagogique de l'Ecole

|  |  |
| --- | --- |
| Formation Théorique | Formation Pratique |
| * Enseignement théorique * Travaux dirigés * Travaux pratiques * Conférences | * Etude de cas * Travaux de réalisation * Projets/ Projets tutoriels * Voyages d’Etudes * Stages en entreprise |

### Architecture des formations pédagogiques

Le recrutement des étudiants à l’ENI se fait uniquement par voie de concours d’envergure nationale en première année de Licence Professionnelle.

Les offres de formation organisées à l’Ecole ont été validées par la Commission Nationale d’Habilitation (CNH) auprès du Ministères de l’Enseignement Supérieur et de la Recherche Scientifique selon les dispositions de l’Arrêté N°31.174/2012-MENS en date du 05 Décembre 2012.

Au sein de l’ENI, il existe une seule mention (INFORMATIQUE) et trois parcours :

* Génie logiciel et Base de Données,
* Administration des Systèmes et Réseaux,
* Informatique Générale.

L’architecture des études à trois niveaux conforment au système Licence-Master-Doctorat (LMD) permet les comparaisons et les équivalences académiques des diplômes au niveau international.

* L =Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6
* M = Master (Bac + 5) = M1, M2 = 4 semesters S7 à S10

Le diplôme de licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtention du diplôme de LICENCE.

Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche d’emploi au terme des études.

Le MASTER RECHERCHE est un diplôme qui remplace l’ancien Diplôme d’Etudes Approfondies (DEA), et qui permet de s’inscrire directement dans une Ecole Doctorale.au terme des études.

* D = Doctorat (Bac +8)
* Le Doctorat est un diplôme qu’on peut obtenir en 3 ans après l’obtention du diplôme de MASTER RECHERCHE.
* Le tableau 2 présente l’architecture des études correspondant au système LMD.



Figure : Architecture des études correspondants au système LMD

La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

Tableau : Liste des formations existantes à l'ENI

|  |  |  |
| --- | --- | --- |
|  | FORMATION | |
| **LICENCE PROFESSIONNELLE** | **MASTER** |
| Condition d’admission | Par voie de concours  Formation Professionnelle : 100 candidats  Formation hybride : 150 candidats |  |
| Condition d’Accès | Bac de série C, D ou Technique | Etre titulaire de licence professionnelle |
| Durée de Formation | 3 ans | 2 ans |
| Diplôme délivré | Diplôme de Licence Professionnelle | Diplôme de Master Professionnel  Diplôme de Master Recherche |

L’accès en première année de MASTER se fait automatiquement pour les étudiants de l’Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s’inscrire directement dans une Ecole Doctorale.

Les Ecoles Doctorales jouissent d’une autonomie de gestion par rapport aux Etablissements de formation universitaire.

Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission Nationale d’habilitation (CNH), l’Ecole Doctorale « Modélisation – Informatique » a été habilitée pour l’Université de Fianarantsoa.

Depuis l’année universitaire 2010-2011, l’ENI s’est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l’insuffisance de la capacité d’accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentielle utilise la visioconférence pour la formation à distance.

Bien qu’il n’existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l’ENI, les étudiants diplômés de l’Ecole sont plutôt bien accueillis dans les instituts universitaires étrangers (Canada, Suisse, France…).

### Relation de l’ENI avec les entreprises et organismes

Les stages effectués chaque année par les étudiants mettent l’Ecole en rapport permanent avec plus de 300 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux.

L’Ecole dispose ainsi d’un réseau d’entreprises, de sociétés et d’organismes publics et privés qui sont des partenaires par l’accueil en stage de ses étudiants, et éventuellement pour le recrutement après l’obtention des diplômes par ces derniers.

Les compétences que l’Ecole cherche à développer chez ses étudiants sont l’adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l’expérimentation et l’innovation.

En effet, la vocation de l’ENI est de former des techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d’évoluer professionnellement dans des secteurs d’activité variés intégrant l’informatique.

Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l’Ecole et les besoins évolutifs du marché de l’emploi. Les principaux débouchés professionnels des diplômés de l’Ecole concernent les domaines suivants :

L’informatique de gestion d’entreprise :

* Les technologies de l’information et de la communication (TIC),
* La sécurité informatique des réseaux,
* L’administration des réseaux et des systèmes,
* Les services bancaires et financiers, notamment le Mobile Banking,
* Les télécommunications et la téléphonie mobile,
* Les Big Data,
* Le commerce, la vente et l’achat, le Marketing,
* L’ingénierie informatique appliquée,
* L’écologie et le développement durable.

Parmi les sociétés, entreprises et organismes partenaires de l’Ecole, on peut citer : ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie ( AUF) , B2B, Banque Centrale, BFG-SG, BIANCO, CHU, BLUELINE, Bureau national de gestion des Risques et des catastrophes (BNGRC), CEDIIFianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CNRIT, COLAS, Direction Générale des Douanes, DLC, FID, FTM, GNOSYS, IBONIA, INGENOSIA, INSTAT, JIRAMA, JOUVE, MADADEV, MANAO, MEF, MEN, MESupRES, MFB, MIC, Min des postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société d’Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, Société LAZAN’I BETSILEO, WWF …

L’organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d’embauche pour les diplômés de l’Ecole.

### Partenariat au niveau international

Entre 1996 et 1999, l’ENI avait bénéficié de l’assistance technique et financière de la Mission Française de Coopération et d’action culturelle dans le cadre du Programme de Renforcement de l’Enseignement Supérieur (PRESUP) consacré à l’Ecole a notamment porté sur :

* Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques :
* La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque ;
* L’appui à la formation des formateurs ;
* L’affectation à l’Ecole d’Assistants techniques français.

De 2000 à 2004, l’ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d’Ingénieurs et Technicien d’Expression Française (CITEF).

Les Enseignants-Chercheurs de l’Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI)

L’ENI avait également signé un accord de coopération interuniversitaire avec l’Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l’Université de la Réunion, l’Université de Rennes 1, l’INSA de Rennes, l’Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l’ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d’accès à Internet de l’Ecole ainsi que de l’Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l’USAID Madagascar, l’ENI de l’Université de Fianarantsoa avait été dotées d’une ligne spécialisée d’accès permanent au réseau Internet.

L’ENI avait de même noué des relations de coopération avec l’Institut de Recherche pour le Développement (IRD).

L’objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu’à Vondrozo (COFAV). Dans ce cadre, un atelier scientifique international avait été organisé à l’ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l’IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar (FPPSM). Des étudiants en DEA et des Doctorants issus de l’ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l’ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l’Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC ;

Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l’Université de Savoie avait été signée par les deux parties concernées. La mise en œuvre de la Convention de 13 Coopération avait permis d’envoyer des étudiants de l’ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l’ENI avait signé en Septembre 2009 un protocole de collaboration scientifique avec l’ESIROI – STIM de l’Université de la Réunion.

Comme l’ENI constitue une pépinière incubatrice de technologie de pointe, d’emplois et d’entreprises, elle peut très bien servir d’instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l’Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sur l’orbite de la modélisation grâce au développement des nouvelles technologies.

### Débouchés professionnels des diplômés

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L’environnement socio-politique du pays depuis 2008 jusqu’ à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants.

Cependant, les formations proposées par l’Ecole permettent aux diplômés d’être immédiatement opérationnels sur le marché du travail avec la connaissance d’un métier complet lié à l’informatique aux TIC.

L’Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante.

Les diplômés en LICENCE et en MASTER issus de l’ENI peuvent faire carrière dans différents secteurs.

L’Ecole bénéficie aujourd’hui de 39 années d’expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C’est une Ecole Supérieure de référence en matière informatique.

Par conséquent, en raison de fait que l’équipe pédagogique de l’Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l’Ecole sont dotés d’une grande expérience dans l’enseignement et dans le milieu professionnel.

L’Ecole est fière de collaborer de façon régulière avec un nombre croissant d’entreprises, de sociétés et d’organismes publics et privés à travers les stages des étudiants. Les formations dispensées à l’Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L’Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L’Ecole s’efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l’informatique de gestion ainsi que l’administration des réseaux et systèmes.

Tableau :Débouchés professionnels des diplômés

|  |  |
| --- | --- |
| LICENCE | * Analyste * Programmeur * Administrateur de Site Web/ de portail Web * Assistant Informatique et Internet * Chef de projet Web ou MultiMedia * Développeur Informatique ou MultiMedia * Intégrateur Web ou Web Designer * Hot Liner/ Hébergeur Internet * Agent de référencement * Technicien/Supérieur de help desk sur Informatique * Responsable de sécurité web * Administrateur de réseau * Administrateur de cybercafé |
| MASTER | * Administrateur de cyber café * Administrateur de réseau et système * Architecture de système d’information * Développeur d’application /web /java/Python/ IOS /Android * Ingénieur réseau * Webmaster /web designer * Concepteur Réalisateur d’applications * Directeur du système de formation * Directeur de projet informatique * Chef de projet informatique * Responsable de sécurité informatique * Consultant fonctionnel ou freelance |

### Ressources humaines

Les ressources humaines affectées à l’Ecole Nationale d’Informatique sont composées de :

* Directeur de l’Ecole : Professeur MAHATODY Thomas, Docteur HDR
* Responsable de Mention : Monsieur RABETAFIKA Louis Haja, Maitre de Conférences
* Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian, Assistant d’Enseignement Supérieur et de Recherche
* Responsable de Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA, Assistant d’Enseignement Supérieur et de Recherche
* Responsable de Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy, Assistant d’Enseignement Supérieur et de Recherche
* Nombre d’Enseignants permanents : 12 dont un (01) Professeur Titulaire, deux (02) Professeurs, Six (06) Maîtres de Conférences, cinq (05) Assistants d’Enseignement Supérieur et de Recherche.
* Nombre d’Enseignants vacataires : 10
* Personnel Administratif : 23

## Chapitre 2 : Présentation de l’établissement d’accueil

* 1. Information d’ordre générale

La douane étant une administration bicentenaire, opte un slogan « Douane, une administration innovante, redevable, partenaire pour l’émergence du pays ». La direction générale de la douane est siégée à l’Immeuble des finances, Rue général Rabehevitra Antaninarenina avec une adresse postal BP 262- Antananarivo Madagascar, une adresse électronique assistance@douanes.mg, et un site web [www.douanes.gov.mg](http://www.douanes.gov.mg)

### Historique

La douane Malagasy est une administration qui a fêté ses 200ème anniversaire en 2020 c’est-à-dire existante depuis 1820. Sa naissance remonte à l’époque de la monarchie royale merina qui est un groupe ethnique issu des « hauteurs » d’origine asiatique assez marqué, résident au centre de l’île du début du 19ème siècle.

L’abolition de l’esclavage Anglo-merina du 23 Octobre 1817 contribue à l’intensification du commerce extérieur sous le règne de Radama I et, par conséquent, à la mise en place de la douane pour alimenter les caisses de l’Etat monarchique. On assiste alors à la création des premières recettes des douanes sur le littoral est, ouest et sud. Des recettes qui auront plusieurs points communs : elles ont été confiées à des parents du roi, elles ont été dirigées par des généraux ce qui montre que la douane était alors un service militarisé, elles ont pour principal objectif d’intensifier le commerce extérieur et de faire rentrer des devises.

Les périodes citées ci-après sont des moments clés qui sont bien à savoir :

* **1828-1861** : Le règne de Ranavalona I, qui est le plus long règne de l’histoire qui a duré 33 ans, il a été caractérisé par 3 périodes :
* 1828-1845 : le prolongement du système adopté sous le règne de Radama I
* 1845-1853 : la fermeture des frontières
* 1853 : réouverture des frontières et le rétablissement du commerce extérieur

Avec la réouverture des ports, les droits de douane furent destinés exclusivement à la cassette royale et étaient perçus à l’import et à l’export de tous les biens. Seules étaient interdites les exportations d’esclaves suivant un traité signé par Radama I. Il est très intéressant de connaître le mode de séparation des Droits de Douane au milieu du 19ème siècle, sans connaître l’esprit : 10/120ème aux proches parents de la reine et à quelques favoris, 11/120ème allaient aux officiers supérieurs, et 99/120ème à la cassette royale.

En Bref, ayant réussi après Radama I à asseoir son autorité sur l’île toute entière, Ranavalona I a su constituer un véritable « corps des douanes » chargé de garder non seulement l’étanchéité de la frontière maritime mais aussi de réglementer les échanges commerciaux qui n’étaient possibles que dans les points de trafic continuellement gardées par les soldats douaniers.

* **1861-1863** : sous Radama II, la douane royage n’a été que le prolongement de celle du régime précédent. Toutefois, par décret du 20 août 1861 ; Radama II supprima les droits de douane à l’importation et à l’exportation ; ce qui causa beaucoup d’amertume chez les bénéficiaires de la répartition et favorisa la spéculation. La politique d’ouverture prônée par Radama II va lui coûter la vie. A la mort du roi, la douane jouait toujours son rôle budgétaire.
* **1863-1868** : sous le règne de Rasoherina, durant ce règne et aussi de Ranavalona II en 1868-1883 les historiens n’ont pu signaler aucun trait particulier de la douane royale. Le système douanier précédent continuait à assurer les rentrées fiscales, nécessitées par le pouvoir monarchique.
* **1883-1896** : sous le règne de Ranavalona II, comme précédemment, la fonction de la douane royale n’a pas évolué quant au système de prélèvement. Toutefois, elle prenait une importance grandissante, du fait de l’ouverture vers l’extérieur. L’Etat monarchique entretenait des relations diplomatiques avec les principaux pays de l’occident (la France, l’Allemagne, les Etats-Unis d’Amérique).

Sous le règne de Ranavalona III, les relations franco-malgaches se sont détériorées, en raison des visées françaises sur certaines parties du territoire national (côte ouest et nord-ouest). La première guerre francohova (Dans sa signification la plus courante à Madagascar même, le terme hova désigne traditionnellement la plus importante subdivision du peuple merina, correspondant aux gens du commun.) qui éclate entre 1883, prend fin décembre1885 par la signature d’un traité assujettissant Madagascar sous le régime du protectorat français.

Le traité contenait deux clauses très importantes. La monarchie cédait deux départements ministériels importants à la France conquérante :

* Les relations avec l’étranger (Ministère des Affaires Etrangères)
* Les recettes douanières

Il était aussi question de répartition des dommages de guerre. Or, les caisses de l’Etat monarchique étaient vides. Un accord de prêt est conclu auprès du Comptoir National d’Escompte de Paris (CNEP). Ce prêt a été garanti par les recettes douanières dont la perception a été confiée au CNEP. Pour assurer ces perceptions, on faisait appel aux expatriés volontaires français (très rares) pour remplacer les douaniers malgaches ; mais, la nécessité de recruter sur place des autochtones s’imposait, pour insuffisance de volontaires expatriés. Les douaniers de cette époque étaient donc recrutés par le CNEP.

* **1896-1960** : sous le régime colonial la douane de Madagascar dépendait étroitement des douanes françaises. L’essentiel des échanges s’effectuE avec la France, étant entendu que le territoire est unique. La souveraineté nationale étant perdue, la douane malgache tombait sous la coupe du régime colonial. De ce fait, la réglementation française était intégralement applicable à Madagascar jusqu’ en 1939. Malgré les difficultés de recrutement du départ, l’effectif douanier était constitué d’expatriés de la Métropole (dirigeant et agents d’exécution). En 1942, la conquête de Madagascar par les anglais suspendait toute activité douanière jusqu’à la fin de la guerre mondiale douanière jusqu’à la fin de la seconde guerre mondiale (1939-1945). En 1942, la protection douanière de la France n’avait plus de sens. Les droits de douane ont été suspendus, mais les taxes locales (TI et TC), ont compensé après leur relèvement, la suppression des droits de douanes.

Ce n’est que très tard après la 2ème guerre mondiale que des malgaches ont été recrutés après la formation à l’école de Myre de Villers pendant deux ans. A la sortie de cette Ecole, ils sont nantis de leur diplôme de CESD (Certificat d’Etudes de Second Degré) équivalant à la classe de 3ème des Etudes Secondaires, formation primaire axée sur le français et l’arithmétique. Le cadre de commis des douanes était créé pour assurer les tâches d’exécution. Le service actif était assuré par le recrutement d’anciens militaires (anciens combattants de la 2ème guerre mondiale) ; il est encadré par des agents métropolitains. La fonction publique était nettement répartie entre deux cadres (métropolitain et indigène). Ce n’est qu’après la Loi Gaston Deferre (loi-cadre) de 1955 que les malgaches ont eu accès aux cadres supérieurs. Le territoire douanier étant confondu avec celui de la France et comme l’essentiel du commerce se faisait avec la France, les réglementations douanières étaient calquées sur celles françaises. Les perceptions effectuées se limitent à deux taxes : la TI et la TC ; la troisième qui est la TT (taxe sur la transaction de 2%) était perçue pour le compte des Impôts Indirects.

* **1963-1974** : sous la première République, la Loi de Finances pour 1961 a rétabli la perception des droits de douane. Le 23 octobre 1963, MADAGASCAR a adhéré au GATT en tant que Partie Contractante. De 1964 à 1974, MADAGASCAR figurait parmi les Etats associés à la CEE et appliquait les règles d’origine de Yaoundé I, Yaoundé II, pour bénéficier du traitement préférentiel dans les échanges avec la CEE. S’agissant d’un régime de libre-échange à double sens, les marchandises originaires de la CEE étaient exonérées de droits de douane mais rentrait à MADAGASCAR dans le cadre du contingent CEE (licence CEE). Inversement, les produits malgaches bénéficient à l’entrée du marché communautaire, du traitement préférentiel (absence de restriction quantitative et exemption de droits de douane). Seules, les marchandises originaires des pays tiers sont frappées des droits de douane (droits protecteurs). Ces droits sont de deux sortes : le tarif général et le tarif minimum. Ce dernier est seul inscrit dans le tarif ; il est perçu sur les marchandises originaires des pays tiers, parties contractantes du GATT. Le tarif général est réservé aux pays tiers non membres du GATT et était le triple du tarif minimum.
* **Deuxième et troisième république** : l’adhésion de Madagascar à des conventions multilatérales et régionales ne fait que compliquer davantage le contrôle douanier.

a) Les diverses conventions de Lomé (I, II, II, IV) et depuis 1995, la Convention de Cotonou.

Il s’agit d’une convention de libre-échange à sens unique : seuls les produits ACP bénéficient de traitement préférentiel. Les droits de douane ont été rétablis vis-à-vis de la CEE depuis 1976.

b) Aménagements régionaux : COI, COMESA, SADC. Ce sont des arrangements régionaux qui ont un double objectif :

* Constituer une zone de négociation dans les enceintes internationales
* But final : Union Douanière pour favoriser l’intégration régionale

**1998-2022** : vers la modernisation

* **1998** : Partenariat public-privé pour le contrôle avant embarquement avec le Bureau Veritas
* **2005** : 1ère stratégie de modernisation : informatisation, refonte complète du Code des Douanes ; Partenariat public-privé avec la société SGS
* **2007** : Lancement de Sydonia++, nouveau système de dédouanement
* **2008** : 2ème stratégie de modernisation, facilitation du commerce, administration phare, administration de confiance.
* **2015** : Mise en place du guichet unique TradeNet
* **2018** : Lancement Sydonia World, en remplacement à Sydonia++
* **2020** : 3ème stratégie de modernisation avec le plan stratégique 2020-2023, Douane, une administration innovante, redevable, partenaire pour l’émergence du pays
* **2021** : Mise en place de la Brigade canine à Antananarivo dans un premier temps (Coopération avec l’Ambassade du Japon et l’OIM)
* **2022** : Effectivité du 100% scanning des conteneurs à l’import et à l’export, Bureau des Douanes Toamasina
* De nos jours : la douane actuelle, La Douane Malagasy s’efforce de perfectionner les moyens matériels et de former un personnel qualifié. Elle fait face aux difficultés engendrées par la mondialisation et la globalisation des échanges en modernisant les moyens destinés à perfectionner les modalités de perception, à assurer le « contrôle a posteriori » et à mieux garder ses frontières mesurant 5 000 km. La Douane d’aujourd’hui se veut être moderne pour mieux conforter ses missions fiscales, économiques et sécuritaires. L’Administration douanière a lancé son plan stratégique pour la période 2020-2023, lequel ambitionne la modernisation de la Douane Malagasy. A cet effet, un programme de modernisation composé de trente-et-un projets est mis en œuvre pour faire de la Douane une administration innovante, redevable, partenaire pour l’émergence du pays.

### Mission

La douane possède des nombreuses valeurs comme l’honneur, la redevabilité, le professionnalisme et le partenariat et adopte principalement quatre grandes missions :

* La sécurisation des recettes douanières ;
* La promotion de la croissance économique facilitant le commerce légitime ;
* La protection des citoyens et de l’environnement en luttant contre les trafics illicites ;
* La sécurisation de la chaîne logistique internationale.

### Organigramme

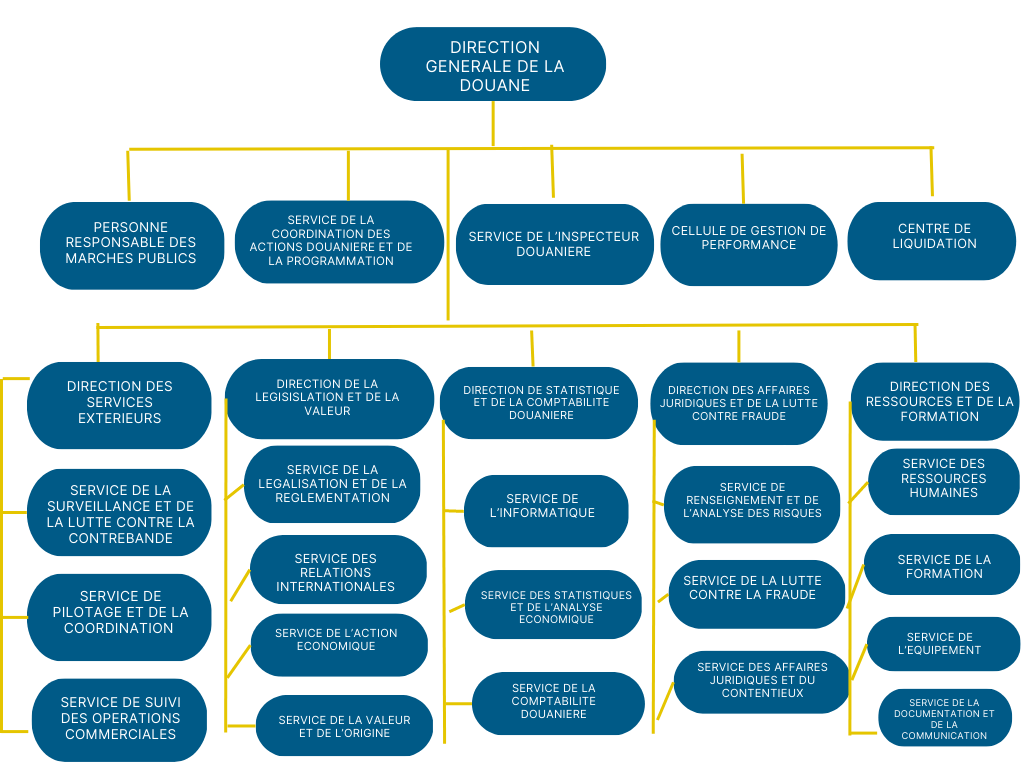
Dans la Figure 4 présente l’organigramme de la direction générale des douanes malagasy :

Figure : Organigramme de la direction générale des douanes malagasy

## Chapitre 3 : Description du projet



### Formulation

Le projet est placé au niveau interne de la douane, facilite les interventions aux problèmes liés au travail de chaque agent. Cela consiste, donc, à mettre en place un système de ticketing à ce niveau. Ce qui fait concevoir et réaliser une application web pour bien gérer les interventions aux problèmes. Cette application sera utile pour chaque employé pour demander de l’aide ou de venir en aide, mais aussi de pouvoir tout de suite voir la solution si le même problème était déjà résolu.

### Objectif et besoins de l’utilisateur

#### Objectif

L’objectif de ce projet est de concevoir et réaliser une application web capable de rendre plus facile les interventions aux problèmes de chaque agent, d’éliminer les pertes de temps à solutionner tout le temps le même problème, de réduire les dépenses d’énergie à se déplacer partout, mais aussi de rendre plus facile la détection des personnes à qui s’adresser lors d’un blocage.

En résumé ce projet vise à :

* Eliminer la redondance de tâches ;
* Economiser les pertes de temps et énergie ;
* Rendre facile la détection des intervenants à qui s’adresser.

#### Besoins de l’utilisateur

Besoins de chaque agent

Les besoins communs de chaque agent sont de :

* + Consulter la liste de catégories et sous-catégories de problèmes ;
  + Voir tous les problèmes existants, et les solutions de ceux qui sont déjà résolus ;
  + Signaler des problèmes.

Besoins des intervenants

Chaque intervenant a besoin de :

* + Voir les problèmes qu’il a reçu ;
  + Envoyer réponse aux problèmes.



### Moyens nécessaires à la réalisation du projet

Des moyens matériels et humains mais aussi logiciels sont nécessaires pour réaliser le projet.



#### Moyens humains

A la réalisation du projet il faudrait des ressources humaines dont à compter :

* Un chef de projet,
* Un concepteur,
* Un développeur.

Les deux derniers peuvent être pris en main par une personne.

#### Moyens matériels

Non seulement des moyens humains seront nécessaire pour ce projet mais aussi des moyens matériels seront utiles tel que :

* Un ordinateur portable de marque HP, de système Windows, de processeur Core i5, de processus 2.4GHz, de RAM 8 Go, et de disque dure 500Go ;
* Une connexion Internet,
* Une source d’électricité,
* Un serveur.

#### Moyens logiciels

Puisqu’on parle de développement d’application, des moyens logiciels seront indispensables comme :

* Un outil de conception ;
* Un système de gestion de base de données (SGBD) ;
* Un environnement de développement (IDE).

### Résultats attendus

On attend à ce que l’application conçue apporte une optimisation au niveau des interventions aux problèmes, qu’elle soit plus performante et facile à manipuler pour chaque agent, qu’elle réponde aux besoins de chaque utilisateur, mais aussi emmène à un nouvel air technologique au niveau de l’entreprise.



#### Chronogramme des activités

Le stage a durée 3mois, le **Erreur ! Source du renvoi introuvable.** montre le planning des actions menées durant ce stage pour atteindre l’objectif.

Tableau : Chronogramme des activités

|  |  |  |
| --- | --- | --- |
| Date Début | Date Fin | Actions |
| 3 octobre 2022 | 13 octobre 2022 | Préparation, accueil de l’entreprise, analyse et collecte des données |
| 14 octobre 2022 | 12 Novembre 2022 | Phase de conception |
| 13 Novembre 2022 | 27 Novembre 2022 | Phase d’apprentissage de la technologie |
| 28 Novembre 2022 | 22 Décembre 2022 | Phase de réalisation |
| 22 Décembre 2022 | 07 Janvier 2023 | Phase de préparation mémoire |

# PARTIE II : ANALYSE ET CONCEPTION

## Chapitre 4 : Analyse préalable

Ce chapitre contiendra : l’analyse du déroulement des interventions aux problèmes à présent, plus des critiques afin d’observer ses atouts et problèmes, puis des ébauches de solutions et la solution retenue qui répond aux besoins de l’utilisateur.



### Analyse de l’existant

L’organisation actuelle et l’inventaire des moyens matériels et humains seront parlés dans cette analyse de l’existant.



#### Organisation actuelle

Actuellement, si un agent affronte un problème, il se met à chercher de l’aide auprès de tierces personnes. Mais il est possible que cette personne ne soit pas l’idéal pour lui venir en aide. Et si cette dernière connaît une personne plus appropriée, elle lui conseillera de la contacter, sinon l’agent doit se débrouiller à chercher une autre personne à qui s’adresser et peut même se déplacer partout si nécessaire.

Au niveau de l’intervenant concerné à un problème, il est possible qu’il résolve un problème déjà résolu plusieurs fois à chaque fois qu’un agent le signale, il est aussi probable qu’il oublie la solution efficace qu’il a déjà proposée auparavant. Des déplacements seront aussi utiles si le problème est un peu vague et difficile à bien comprendre.

Donc on peut constater que les interventions au problème actuel se comporte :

* Oralement et à l’écoute,
* A essayer de trouver à qui demander de l’aide à chaque problème,
* A se déplacer partout si nécessaire,
* A utiliser le cerveau pour la mémorisation des solutions efficaces ;
* A solutionner tout le temps des problèmes identiques.

#### Inventaire des moyens matériels et logiciels

##### Moyens matériels

Tous les agents de la douane Malagasy disposent tous d’un ordinateur pour effectuer leurs tâches, certains optent d’un ordinateur de bureau, les autres des ordinateurs portables qui ont tous accès à une connexion internet. La douane dispose aussi d’un serveur.

Le Tableau 5 représente les caractéristiques de chaque ordinateur.

Tableau : Moyens matériels

|  |  |
| --- | --- |
| Matériels | Configurations |
| Ordinateur de bureau | Processeur Intel® Core™ i5; Processeur Intel® Core™ i3; 32 Go de mémoire SDRAM DDR4 ,windows 10 |
| Ordinateur portable | HP Intel® Core™ i5- 4210U CPU@ 1.70GHz 2.40GHz(4CPUs),4G de RAM, Windows 10, 500Go de disque dure |

##### Moyens logiciels

La douane Malagasy, possède déjà plusieurs logiciels comme :

* Environnement de travail : des systèmes d’exploitation Windows ou Linux dépend du choix des agent,
* IDE ; NetBeans, Eclipse, VS Code,
* Un serveur et SGBD : ORACLE.

### Critique de l’existant

Après l’analyse précèdent, plusieurs problèmes peuvent être dégagés :

* Risque de se déplacer partout,
* Perte de temps et d’énergie au niveau de la personne ayant le problème même et aussi de l’intervenant,
* Redondance de tâches à faire,
* Perte de données utiles, et possibilité de mélanger des informations,
* Difficulté de résoudre un problème ;

### Conception avant-projet



#### Proposition des solutions

Trois solutions pourront être envisagées pour ce projet :

* Planifier régulièrement des formations et moment de partage pour tous les agents de la douane,
* Créer un poste spécial, qui se charge d’assistance d’aide,
* Concevoir et réaliser une application pour gérer les interventions aux problèmes Tableau : Comparaison entre les solutions

|  |  |  |
| --- | --- | --- |
| SOLUTION | AVANTAGES | INCONVENIENTS |
| * Planification des formations et moment de partage | * Beaucoup d’ébauche de sujets intéressants et nécessaires pour résoudre un problème. * Moment de cohésion entre chaque agent * Interaction directe apportant plus de compréhension | * Déplacement * Trop de temps à consacrer * Dépense financière et d’énergie * Manque de concentration * Perte de données |
| * Création d’un poste spécial assistance d’aide | * Pas besoin de chercher partout où demander de l’aide * Problème et solution unifiés en un endroit * Création de nouveau emploi | * Difficulté de trouver le spécialiste idéal * Trop de démarche à suivre pour l’implémentation * Trop de problèmes en queue * Trop de travail à faire |
| * Conception et réalisation d’une application capable de gérer les interventions aux problèmes | * Facilité de résoudre un problème * Adapter aux matériels et logiciels existants * Stockage de donnée plus fiable * Diminue le déplacement * Existence d’une division qui s’en charge des autres applications | * Dépense de temps pour la réalisation * Nécessite aux moins deux personnes * Demande beaucoup d’études et d’analyses |

Solution retenue :

La dernière solution a été prise en compte comme solution adéquate et avantageuse car elle répond plus aux besoins de l’utilisateur et montre moins de désavantages comparer aux deux autres.

#### Choix d’outils pour la conception et la réalisation

##### Méthode de conception

Parmi les méthodes approfondies durant la formation théorique, on peut comparer Merise et Unified Process. Le Tableau 7 montre la comparaison entre eux :

Tableau : Comparaison entre MERISE et Unified process

|  |  |  |
| --- | --- | --- |
|  | **MERISE** | **UNIFIED PROCESS** |
| **Avantages** | * Séparation des traitements et données * Les niveaux sont bien précis et ordonnés | * Orientée objet * Adopte des caractéristiques : itératif, incrémental piloté par les risques, Orienté composant, et orienté utilisateur * Utilise des langages de modélisation |
| **Inconvénients** | * Avec étapes qui doivent être suivies sans en manquer un pour obtenir un bon résultat * Un peu éloigné du langage et difficile d’implémenter | * Avec séparation des cycles d’abstraction pas très bien précise |

Solution retenue : La méthode 2TUP sera prise en compte comme méthode de conception, grâce aux avantages très impressionnants qu’elle apporte.

 Présentation 2TUP :

Par définition, un processus est une séquence d’étapes en partie ordonnée, Un processus unifié est une méthode de développement logiciel orienté objet, et 2TUP connu par le nom de 2 Track Unified Process est un processus de développement logiciel qui le met en œuvre. Ce dernier est basé sur deux axes principaux : l’axe fonctionnelle, qui étudie l’application et l’axe technique, qui étudie l’implémentation. Il est présenté par un cycle en Y, qui montre une étude parallèle entre la branche fonctionnelle et technique. Il s’accomplit en 4 phases :

* Phase fonctionnelle : envisage les fonctionnalités du futur système logiciel,
* Phase technique : développe l’architecture technique et réalise les fonctions les plus prioritaires,
* Phase de conception et réalisation : live toutes les fonctions du système,
* Phase de test : permet de corriger et de faire évoluer le système ;

La Figure 5 présente le cycle en Y de 2TUP [9]

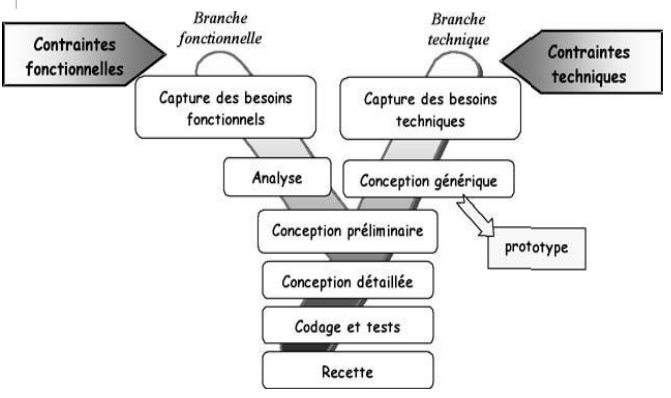


Figure : Le cycle en Y de 2TUP

**Branche fonctionnelle :**

1. Les étapes de la branche fonctionnelle se présentent comme suit :

* L’étape de capture des besoins fonctionnels : produit le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie, au plus tôt, le risque de produire un système inadapté aux utilisateurs.

Cette phase a pour objectif de définir :

* La frontière fonctionnelle entre le système considéré comme une boîte noire et son environnement, c'est le niveau contexte ;
* Les activités attendues des différents utilisateurs par rapport au système toujours envisagé comme une boite noire, c'est le niveau cas d'utilisation.

1. L'étape d'analyse consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en termes de métier.

**Branche technique :**

Les étapes de la branche technique se présentent comme suit :

1. L'étape de capture des besoins techniques recense toutes les contraintes sur les choix de dimensionnement et la conception du système, les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique). Cette étape permet de définir le modèle d'analyse technique. Le rôle de ce dernier est d'établir les couches logicielles et y spécifier les activités techniques attendues ;
2. L'étape de conception générique définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique ou design pattern (aspect qui sera développé ultérieurement) qui définit les Framework. Ces derniers, délivrant les services techniques, assurent la réponse aux exigences opérationnelles du système

**Branche conception - réalisation**

Les étapes de cette branche se présentent comme suit :

1. L'étape de conception préliminaire est une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer. Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels. Ce modèle regroupe les informations des branches technique et fonctionnelle ;
2. L'étape de conception détaillée permet d'étudier comment réaliser chaque composant. Cette étape produit le modèle de conception des composants. Ce modèle fournit l'image prête à fabriquer du système complet. C'est dans l'étape de codage que s'effectue la production des composants et les tests des unités de code au fur et à mesure de leur réalisation. L'étape de recette consiste à valider les fonctionnalités du système développé.

##### Outil de modélisation

En termes de modélisation, un outil sera nécessaire pour le faire. D’où le tableau 6 permet de comparer deux outils de modélisation.

Tableau : Comparaison de STARUML et VISUAL PARADIGM

|  |  |  |
| --- | --- | --- |
|  | STARUML | VISUAL PARADIGM |
| Avantages | * Disponible à open source * Permet de modéliser une dizaine types de diagrammes * Apporte une facilité d’accès * Interface graphique ergonomique | * Multiplateforme * Disposant de différentes fonctions facilitant la conception des diagrammes aux débutants * Permet de modéliser presque tous les types de diagramme en UML * Génère une code source des diagrammes |
| Inconvénients | * Disponible que sous Windows * Importation des ressources pas assez parfaite | * Pas d’interopérabilité avec autres outils |

Solution Retenue :

Le VISUAL PARADIGM sera utilisé comme outil de modélisation nécessaire pour la réalisation de ce projet.

##### Choix du système de gestion de base de données

Un SGBD est un logiciel qui assure le stockage et partage des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Tableau : comparaison entre deux SGBD existants

|  |  |  |
| --- | --- | --- |
|  | **ORACLE** | **MySQL** |
| **Avantages** | * Assure une sécurisation parfaite * Plus rapide * Plus performant pour des données assez lourdes * Utiliser par l’entreprise à présent | * Open Source * Multiplateforme * Avec architecture Client/serveur |
| **Inconvénients** | * Très coûteux * Assez compliqué | * Ne supporte pas de donnée de grande taille * Pas assez performant |

Solution retenue :

Grâce aux avantages qu’oracle adopte, il est choisi comme SGBD idéal pour ce projet.

* **Présentation oracle**

Oracle est un SGBDR multiplateforme, supporte certain nombre de programmations surtout la programmation en JAVA, grâce au grand nombre d’outils qu’il dispose. De nos jours, il est orienté cloud : 12c, 18c,19c,21c.

Sur le terme de fonctionnalité oracle dispose :

* Une structure de données logique : pour stocker les données afin de pouvoir interagir avec la base de données sans savoir où sont stockées les données physiquement,
* Un partitionnement : permettant de diviser une grande table en différents éléments et de stocker chaque élément sur des périphériques de stockage.
* Une mise en cache de la mémoire : permettant de faire évoluer une très grande base de données qui peut toujours fonctionner à grande vitesse.
* Une Dictionnaire de données : un ensemble de tables et de vues internes permettant d'administrer plus efficacement la base de données Oracle.
* Sauvegarde et restauration : garantissant l'intégrité des données en cas de panne du système, grâce à un outil puissant RMAN pour effectuer des sauvegardes de base de données à froid, à chaud et incrémentielles et des récupérations ponctuelles.
* Un Clustering (ou RAC) : permettant une haute disponibilité qui permet au système de fonctionner sans interruption des services en cas de défaillance d'un ou plusieurs serveurs d'un cluster.

Oracle fournit trois éditions principales :

* Enterprise Edition (EE) : courante et coûteuse, ne prend pas en compte le nombre maximum de processeurs, et aussi la limite de mémoire ou de taille de base de données, contient des fonctionnalités premium qui ne sont pas disponibles dans d'autres éditions.
* Edition standard (SE) : limitée de l'édition entreprise, accepte quatre processeurs ou moins, pas limite de mémoire ou de taille de base de données, contient de nombreuses fonctionnalités, mais pas autant que EE.
* Expression Edition (XE) : version gratuite de la base de données Oracle disponible sur les plates-formes Windows et GNU/Linux, accepte seulement 2 processeurs, et au maximum 2 Go de RAM plus de 12 Go de données utilisateur, adopte une fonctionnalité très limitée.

##### Langage de programmation

Le langage de programmation joue un rôle majeur pour la formulation et réalisation d’une application. Le tableau 8 montre la comparaison entre deux langages de programmation.

Tableau : Comparaison entre PHP et JAVA

|  |  |  |
| --- | --- | --- |
|  | **PHP** | **Java** |
| **Avantages** | - De communauté dynamique  - Possède beaucoup de documentation  - Multiplateforme  - Open Source  - Simple à apprendre | - Multiplateforme  - De communauté dynamique  - Orienté Objet  - Adapté pour toutes types d’application (web, desktop, mobile)  - Possède beaucoup de documentation  - Conforme pour l’entreprise |
| **Inconvénients** | - Spécifique pour l’application web  - Débogage assez fastidieux | - Nécessitant un serveur d’application pour une application web  - Prenant plus de ressources |

Solution retenue :

La meilleure solution retenue au niveau du choix du langage de programmation est de s’orienter vers Java. En profitant du Framework java « Spring Boot ».

##### Environnement de développement (IDE)

Pour le développement il faudra, un outil de développement web. Le Tableau 11 : Comparaison les deux outils utiles pour le développement Web sera utile pour comparer deux outils utiles pour le développement web.

Tableau : Comparaison les deux outils utiles pour le développement Web

|  |  |  |
| --- | --- | --- |
|  | **Visual Studio Code** | **NetBeans** |
| **Avantages** | * Livré avec un support intégré pour TypeScript. * Avec licence open source * Développé par Microsoft * Suffit juste de télécharger des extensions pour installer Spring Boot | * Avec licence open source * Avec possibilité de déploiement des applications web vers Tomcat avec le « pack web » |
| **Inconvénients** | * Assez récent | * Avec manque de quelques fonctionnalités |

Solution retenue :

La solution retenue est de servir le Visual Studio Code comme environnement de développement.

## Chapitre 5 : Analyse et Conception

Pour concevoir un modèle conceptuel, il faut tout d’abord établir le dictionnaire des données. A ce propos, un dictionnaire des données est l’ensemble de toutes les informations collectées et qui participent à la conception du Système d’Information.



### Dictionnaire des données

Un dictionnaire des données est constitué par :

* Le nom ou code ou identification de la rubrique,
* La définition ou présentation ou même description de cette rubrique,
* Sa caractéristique ou structure décrivant ses propriétés : type et taille ;

Tableau : Le dictionnaire des Données

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom du rubrique | Description | Caractéristiques | | Remarque |
| Type | Taille |  |
| ConfidentialiteProb | Confidentialité du problème | L |  | Public/ Privé |
| CPVille | Code postal de la ville | AN | 3 |  |
| dateDebCat | Date début de la catégorie | D | 12 | jj /mm/aaaa hh :mm |
| dateDebHier | Date début de la hiérarchie | D | 12 | jj /mm/aaaa hh :mm |
| dateDebPoste | Date début du poste | D | 12 | jj /mm/aaaa hh :mm |
| dateEnvProb | Date d’envoi d’un problème | D | 12 | jj /mm/aaaa hh :mm |
| dateEnvRep | Date d’envoi de la réponse | D | 12 | jj /mm/aaaa hh :mm |
| dateFinCat | Date fin de la catégorie | D | 12 | jj /mm/aaaa hh :mm |
| dateFinHier | Date fin de la hiérarchie | D | 12 | jj /mm/aaaa hh :mm |
| dateFinPoste | Date fin du poste | D | 12 | jj /mm/aaaa hh :mm |
| descriptionCat | Description de la catégorie de problème | AN | 500 |  |
| descriptionProb | Description du problème | AN | 500 |  |
| DescriptionRep | Description de la réponse de l’agent | AN | 500 |  |
| emailAgent | Adresse email de l’agent | AN | 50 |  |
| FonctionPoste | Fonction dans un poste | AN | 50 |  |
| idAgent | Identification de l’agent de la douane | N | 5 |  |
| idCat | Identification de la catégorie de problème | N | 5 |  |
| idHier | Identification de la hiérarchie | N | 5 |  |
| idPoste | Identification du poste | N | 5 |  |
| idPrio | Identification de la priorité de problème | N | 5 |  |
| idProb | Identification du problème | N | 5 |  |
| idQuartier | Identification du quartier | N | 5 |  |
| idRep | Identification de la réponse de l’agent | N | 5 |  |
| idRole | Identification du rôle de l’agent | N | 5 |  |
| idStatut | Identification du statut des problèmes | N | 5 |  |
| idType | Identification du type d’hiérarchie | N | 5 |  |
| idVille | Identification de la ville | N | 5 |  |
| libelleCat | Libellé de la catégorie de problème | A | 50 |  |
| libelleHier | Libellé de la hiérarchie | A | 30 |  |
| libellePrio | Libellé de la priorité de problème | A | 20 |  |
| libelleQuartier | Libellé du quartier | A | 50 |  |
| libelleRep | Libellé de la réponse de l’agent | A | 50 |  |
| libelleRole | Libellé du rôle | A | 20 |  |
| libelleStatut | Libellé statut des problèmes | A | 20 |  |
| libelleType | Libellé du type d’hiérarchie | A | 20 |  |
| libelleVille | Libellé de la ville | A | 30 |  |
| libellleProb | Libellé du problème | A | 50 |  |
| mdpAgent | Mot de passe de l’agent | A | 20 | Crypté |
| nomAgent | Nom de l’agent | A | 50 |  |
| numMatAgent | Numéro de matricule de l’agent | AN | 6 |  |
| photoAgent | Photo de l’agent | B |  |  |
| pieceProb | Pièce jointe associée au problème | B |  |  |
| pieceRep | Pièce jointe associée à réponse de l’agent | B |  |  |
| portePoste | Porte d’un poste | AN | 20 |  |
| prenomAgent | Prénom de l’agent | A | 30 |  |
| telAgent | Numéro téléphone de l’agent | AN | 14 |  |

**Légendes :**

A : Alphabétique, AN : Alphanumérique N : Numérique, D : Date, L : Logique, B : BLOB;

### Règles de gestion

Une règle de gestion permet de régir le fonctionnement global d’une application et d’établir les liens entre les données utilisées.

**Règles de gestion relatives aux agents**

**RG01** : Le numéro matricule et l’adresse email d’un agent sont uniques

**RG02** : Un agent peut signaler aucun ou plusieurs problèmes, il pourrait voir automatiquement l’agent à qui demander de l’aide.

**RG03** : Un agent peut envoyer aucune ou plusieurs réponses

**RG04** : Un agent peut recevoir aucun ou plusieurs problèmes venant d’un autre agent

**RG05** : Un agent est rattaché à un poste dans une période bien définie

**RG06** : Un agent peut éventuellement consulter la liste des catégories et sous catégories de problèmes existante, voir la liste de tous les problèmes déjà déclarés et aussi voir la solution des problèmes qui sont déjà résolus.

**Règles de gestion relatives aux problèmes**

**RG07** : Un problème est signalé par un et un seul agent

**RG08** : Un problème est reçu par un ou plusieurs agents

**RG09** : Un problème peut appartenir à aucun ou plusieurs transferts

**RG10** : Un transfert appartient à un et un seul problème

**RG11** : Un problème est associé à un et un seul statut

**RG12** : Un problème est caractérisé par une priorité : urgent, moyen ou faible

**RG13** : Un problème appartient à une catégorie

**Règles de gestion relatives aux réponses**

**RG14** : Une réponse est envoyée par un et un seul agent

**RG15** : Une réponse est envoyée par l’agent seulement qu’après avoir reçu un problème, ou après avoir reçu une réponse si cette dernière n’était pas efficace pour la résolution du problème

**Règles de gestion relatives aux catégories de problème**

**RG16** : Une catégorie appartient à un ou plusieurs problèmes

**RG17** : Une catégorie est constituée par un ou plusieurs sous-catégories

**RG18** : Une sous-catégorie est un constituant d’une seule catégorie

**RG19** : Une catégorie appartient à un ou plusieurs problèmes

**RG20** : Une catégorie appartient à un ou plusieurs problèmes

**RG21** : Une catégorie est rattachée à un poste dans une période donnée

**RG22** : Une catégorie peut être modifiée en un temps donné ou ne pas exister donc, elle est délimitée entre sa date début et date fin afin de mieux détecter sa présence.

**Règles de gestion relatives aux statuts de problème**

**RG23** : Un statut peut être soit nouveau, en attente ou résolu et associé à aucun ou plusieurs problèmes

**RG24** : Dès qu’un problème est signalé le statut du problème est « nouveau », lorsqu’un agent répond à ce problème, sa valeur change en « en attente » et y reste jusqu’à ce qu’il soit défini comme résolu, il change en « résolu », lorsque l’agent valide que la réponse était efficace.

**Règles de gestion relatives aux hiérarchies :**

**RG25** : Une hiérarchie à un seul type

**RG26** : Une hiérarchie est enchaînée par aucune ou plusieurs enchaînements

**RG27** : Un enchaînement est lié à une hiérarchie

**RG28** : Une hiérarchie est délimitée dans une période donnée donc possède une date début et date fin

**Règle de gestion relative aux types d’hiérarchie :**

**RG29** : Un type appartient à un ou plusieurs hiérarchies

**Règles de gestion relatives aux postes :**

**RG30** : Un poste est dans une hiérarchie

**RG31** : Un poste est situé dans un quartier

**RG32** : Un poste est situé dans un quartier

**RG33** : Un poste est délimité par une date début et date fin

**Règles de gestion relatives aux Villes et Quartiers :**

**RG34** : Une ville ne comporte aucun ou plusieurs quartiers

**RG35** : Un quartier est relié à une ville

**RG36** : Un quartier peut contenir aucun ou plusieurs postes

**Règles de gestion relatives aux priorités :**

**RG37** : Une priorité peut caractériser aucun ou plusieurs problèmes

**RG38** : Une priorité peut prendre la valeur urgent, ou normal, ou faible



### Représentation et spécification des besoins

#### Diagramme de cas d’utilisation

##### Concept du diagramme de cas d’utilisation

Le diagramme de cas d’utilisation appartient aux diagrammes comportementaux, permet de décrire les fonctionnalités d’un système d’un point de vue utilisateur, sous la forme d’actions et de réactions. Un diagramme de cas d’utilisation est constitué par :

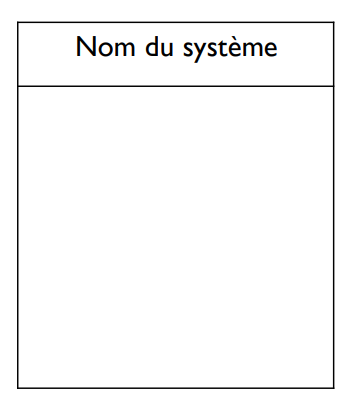
* Un système permet de délimiter le sujet d’étude et ne contient que des cas d’utilisations non des acteurs. Il est représenté par un rectangle contenant le nom du système. La Figure 6 représente le système dans un cas d’utilisation

Figure : Système dans un cas d’utilisation

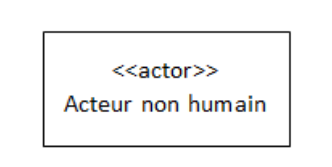
* Un acteur représente un rôle d'un utilisateur qui interagit avec le système, peut être humain ou non humain. Les deux figures montrent la représentation de ces deux types d’acteurs

Figure : Acteur humain et non humain

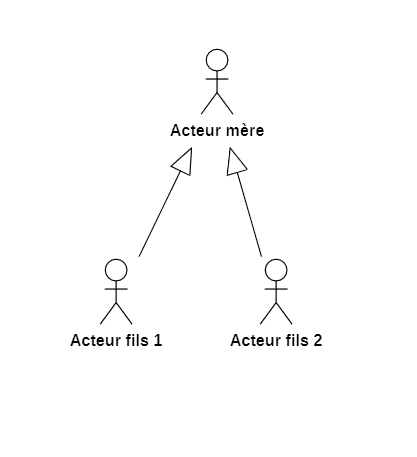
* Relation entre acteur : la relation de généralisation et spécialisation est utilisée pour relier deux acteurs c’est-à-dire l’un hérite de l’autre ou plus précisément : un acteur est fils d’un acteur mère. La représentation de la relation entre acteur est figurée dans la Figure 8

Figure : La relation entre acteur

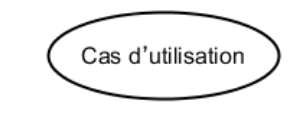
* Cas d'utilisation : décrit une fonction qu'un système exécute pour atteindre l'objectif de l'utilisateur, il est représenté par un ovale contenant le nom de l’action. La Figure 9 montre la représentation d’un cas d’utilisation :

Figure : Cas d'utilisation

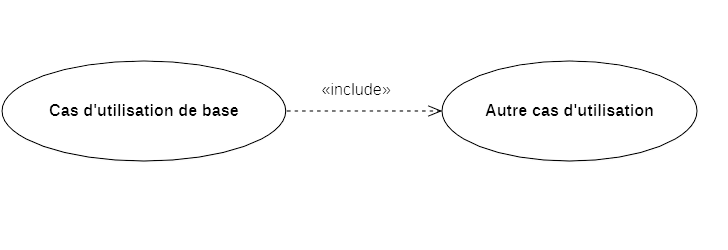
* Relation entre cas d’utilisation : Il existe deux types de relation entre cas d’utilisation : la relation d’inclusion et d’extension
* Relation d’inclusion : connu par le nom « include », qui montre qu’un cas d’utilisation de base insère explicitement un autre cas d’utilisation de manière obligatoire. La figure 9 montre la représentation d’une relation d’inclusion.

Figure : la relation d'inclusion

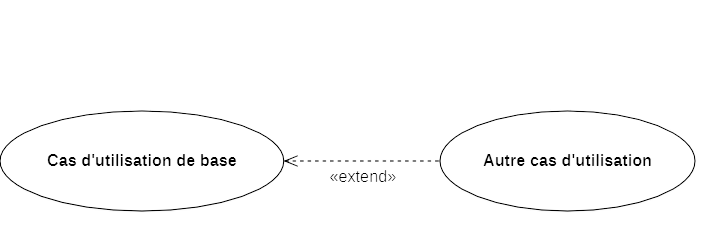
* Relation d’extension : connu par le nom « extend », qui montre qu’un cas d’utilisation de base insère implicitement un autre cas d’utilisation de manière optionnel. La Figure 11 montre la représentation d’une relation d’extension.

Figure :La relation d'extension

##### Recueil des besoins

###### Identification des acteurs :

Le Tableau 13 recense les acteurs identifiés avec leurs rôles respectifs.

Tableau : Identification des acteurs

|  |  |  |
| --- | --- | --- |
| Acteur | Description | Rôle |
| Agent | Employé de la douane | * Consulter catégorie et les sous-catégories de problèmes * Consulter tous les problèmes résolus et non résolus * Envoyer des problèmes * Gérer son profil * S’authentifier * Envoyer des réponses aux problèmes * Transférer les problèmes |
| Dispatcheur | Un agent qui assure le fonctionnement de l’intervention lié à sa direction ou service | * Gérer les catégories de problèmes et ses sous-catégories * Gérer la fiche de postes |
| Administrateur | Agent de la douane qui s’occupe de l'administration fonctionnelle de l’application | * Gérer les hiérarchies existantes dans la douane * Gérer le poste * Gérer les agents |

Identification des messages

Le Tableau 14 représente les messages identifiés pour chaque agent.

Tableau : Identification des messages

|  |  |  |
| --- | --- | --- |
| Acteur | Messages émis | Messages reçus |
| Agent | Demande des catégories de problèmes et sous-catégories de problèmes | Liste des catégories de problèmes |
| Agent | Demande des problèmes déjà existants | Liste des problèmes résolus et non résolus |
| Agent | Envoyer un problème | Liste problème envoyé |
| Agent | Modifier profil | Profil mise à jour |
| Agent | Envoyer une réponse | Liste réponse envoyé |
| Agent | Transférer des problèmes | Liste problème transférer |
| Dispatcheur | Ajout, modification, recherche des catégories et sous-catégories de problèmes | * Liste des catégories et sous catégories * Mise à jour de la liste des catégories et sous-catégorie |
| Dispatcheur | Modification, recherche à la fiche de poste de chaque utilisateur | * Liste de fiche de poste mise à jour |
| Administrateur | Ajout, modification, recherche des hiérarchies et leurs types | * Liste des hiérarchies et types * Mise à jour de la liste |
| Administrateur | Ajout, modification, recherche des postes | * Liste des postes * Mise à jour de la liste |
| Administrateur | Ajout, modification agent | Liste des agent et mise à jour des agents |

###### Modélisation du contexte

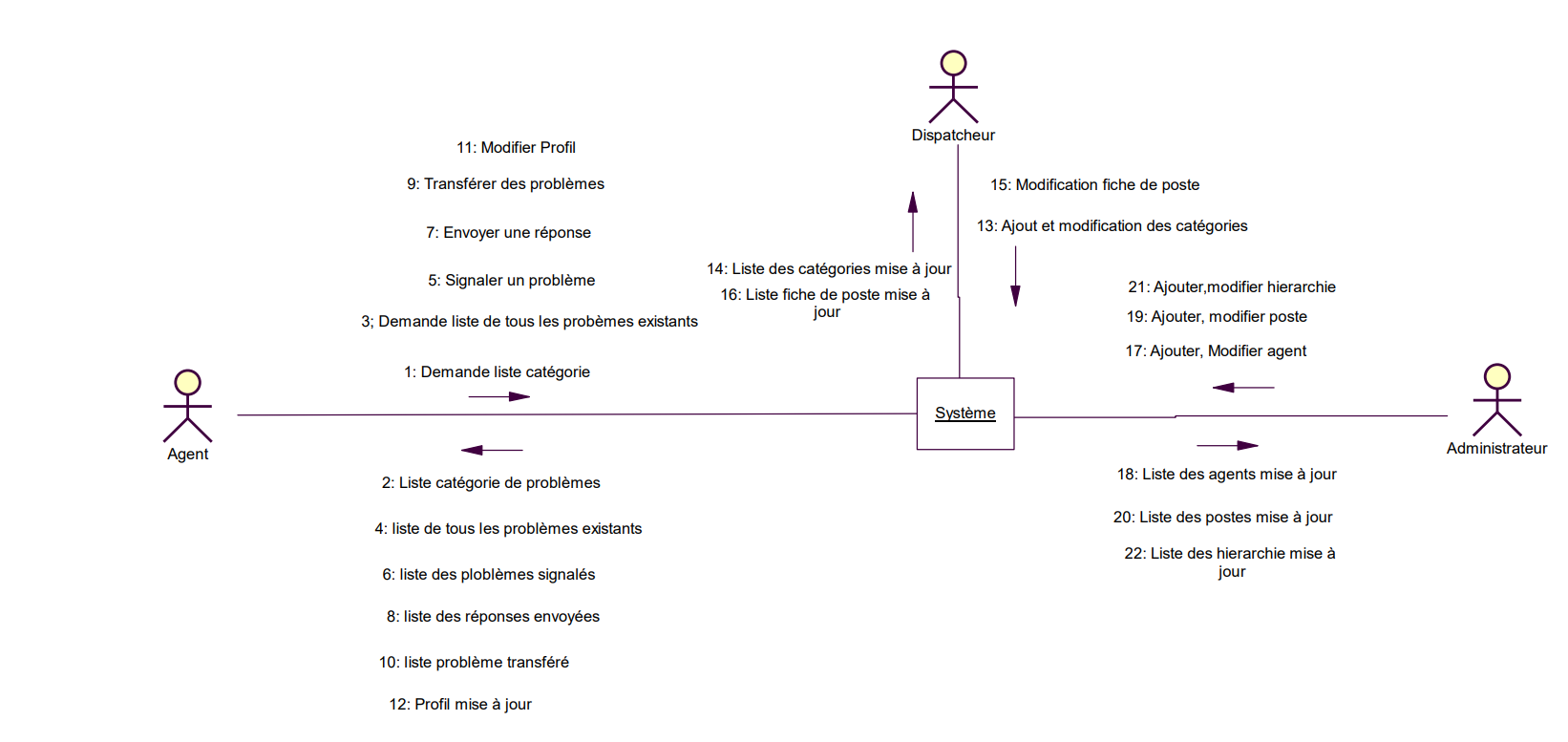
 Le diagramme de contexte permet de montrer les échanges d'informations qui sont réalisés entre notre système et les éléments extérieurs au système.

Figure : Diagramme de contexte



##### Capture des besoins

a) *Identification des cas d’utilisation*

Un cas d'utilisation dis « use case » en anglais décrit les fonctions générales et la portée d’un système.

Le Tableau 15 représente les cas d’utilisations identifiés dans le système

Tableau : Description des cas d'utilisation du système

|  |  |
| --- | --- |
| Fonctionnalité | Description |
| Consulter liste catégorie et sous-catégorie | Permet à chaque agent de voir la liste des catégories et sous-catégories de problèmes |
| Consulter liste de tous les problèmes | Permet à chaque agent d’apercevoir la liste des problèmes existants et de pouvoir visualiser ceux qui sont déjà résolus |
| Gérer profil | Permet aux agents de personnaliser son profil |
| Signaler problème | Permet à chaque agent de pouvoir envoyer des problèmes |
| Envoyer une réponse | Permet à chaque agent de répondre à des problèmes ou répondre à des réponses liées à un problème |
| Transférer un problème | Permet à un agent de transmettre des problèmes à la personne plus appropriée |
| Gérer catégorie et sous-catégorie | Permet à un agent dispatcheur de pouvoir mettre à jour les catégories et sous-catégories de problèmes |
| Gérer fiche de poste | Permet à un agent dispatcheur de mettre à jour le fiche de poste des agents |
| Gérer Agent | Permet à un agent administrateur de mettre à jour la liste des agents |
| Gérer Hiérarchie | Permet à un agent administrateur de mettre à jour les hiérarchies existants dans la douane |
| Gérer poste | Permet à un agent administrateur de mette à jour les postes existants dans la douane |

b*) Diagramme de cas d’utilisation du système*

La phase de recueil et la capture des besoins mènent à pouvoir établir un diagramme de cas d’utilisation du système. La Figure 15 représente le diagramme de cas d’utilisation du système.



Figure : Diagramme de cas d'utilisation du système

#### Priorisation des cas d’utilisation

La priorisation des cas d’utilisations mène à ordonner les actions qui va se passer en priorité, et de bien gérer la fonctionnalité de l’application. Il existe que des cas d’utilisation ne puissent se réaliser qu’après un autre cas d’utilisation, donc c’est ceux dont nous allons pouvoir visualiser ici, le Tableau 16 représente la priorisation des cas d’utilisation du système.

Tableau : Priorisation des cas d'utilisation du système

|  |  |  |
| --- | --- | --- |
|  | Cas d’utilisation | Priorité |
| BackOffice | S’authentifier | 1 |
| Gérer Hiérarchie | 2 |
| Gérer Poste | 3 |
| Gérer Agent | 3 |
| Gérer catégorie et sous-catégorie | 3 |
| Gérer Fiche de poste | 3 |
| FrontOffice | S’authentifier | 1 |
| Consulter catégorie et sous-catégorie de problème | 2 |
| Consulter tous les problèmes existants | 2 |
| Gérer Profil | 2 |
| Signaler un problème | 3 |
| Envoyer une réponse | 3 |
| Transférer un problème | 3 |

#### Spécification des besoins techniques

Le SBT tourne autour des contraintes et choix qui mesurent la conception du système. Il parle de la partie non fonctionnelle d’une application, c’est-à-dire d’un côté plutôt technique.

Les principaux besoins non fonctionnels pour cette application sont :

* La rapidité de traitement : vise à constater que l'exécution des traitements s'approche le plus possible du temps réel ;
* La performance : favorise à répondre à toutes les exigences des usagers d'une manière optimale ;
* La convivialité : montre une simplicité à la manipulation ;
* La sécurité : met en sorte que l’application soit protégée par un système d’authentification.

### Modélisation du domaine

#### Diagramme de séquence système pour chaque cas d’utilisation

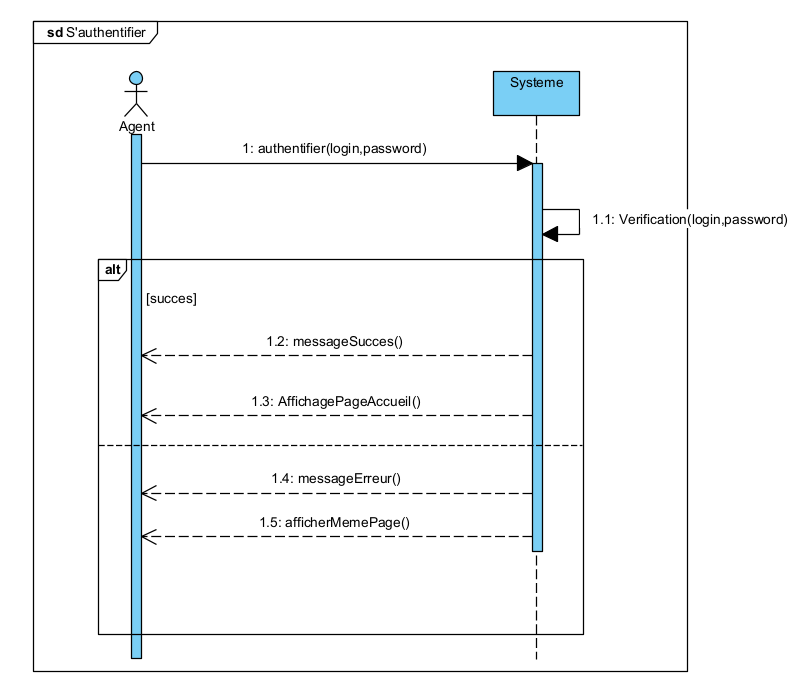
 Le diagramme de séquence montre l’interaction entre objet, elle représente comment et dans quel ordre des objets fonctionnent ensemble. Ici on va voir tous les diagrammes de séquences de chaque cas d’utilisation.

Figure : Diagramme de séquence "S'authentifier"

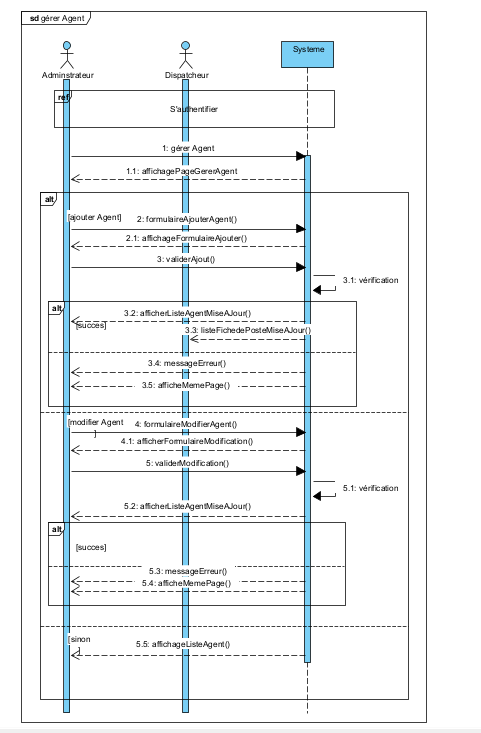
La Figure 15 représente le diagramme de séquence du cas d’utilisation « Gérer Agent ».

Figure : Diagramme de séquence : "Gérer Agent"

La Figure 16 représente le diagramme de séquence du cas d’utilisation « Gérer Hiérarchie »

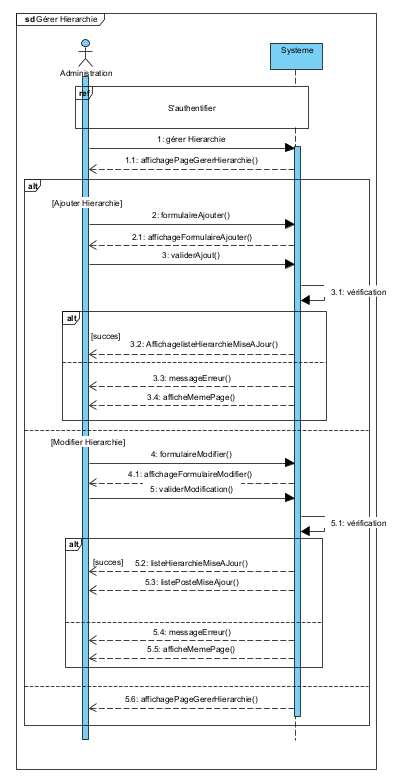


Figure : Diagramme de séquence : "Gérer Hiérarchie"

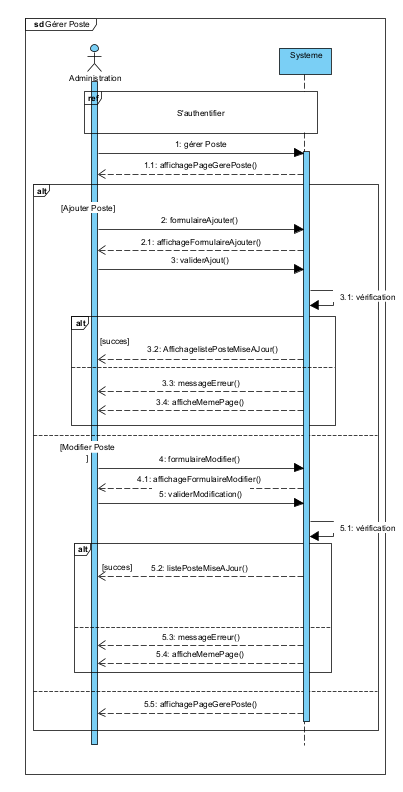
Le diagramme de séquence du cas d’utilisation « Gérer Poste » est figuré dans la Figure 17 .

Figure : Diagramme de séquence "Gérer Poste"

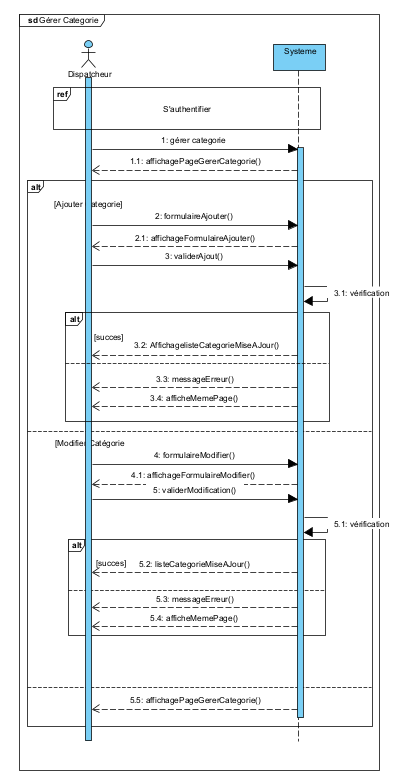
La Figure 18 représente le digramme de séquence au niveau du cas d’utilisation « Gérer Catégorie et sous-catégorie ».

Figure : Diagramme de séquence "Gérer Catégorie"

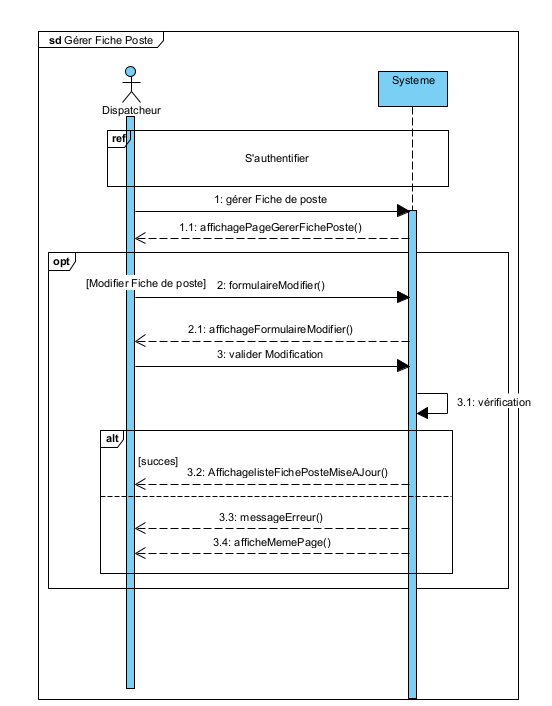
****La Figure 19 représente le diagramme de séquence du cas d’utilisation « Gérer Fiche de Poste »

Figure : Diagramme de séquence "Gérer Fiche Poste"

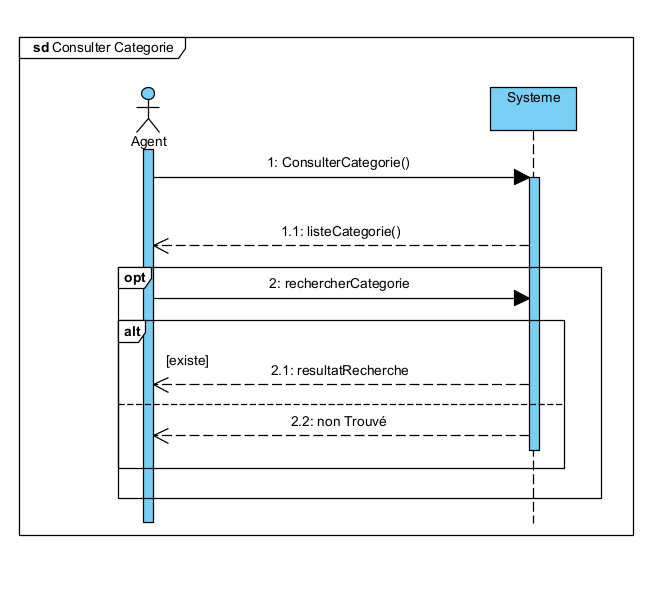
Le diagramme de séquence au niveau du cas d’utilisation « Consulter catégorie et sous-catégorie » est représenté dans la Figure 22.

Figure : Diagramme de séquence "Consulter Liste Catégorie"

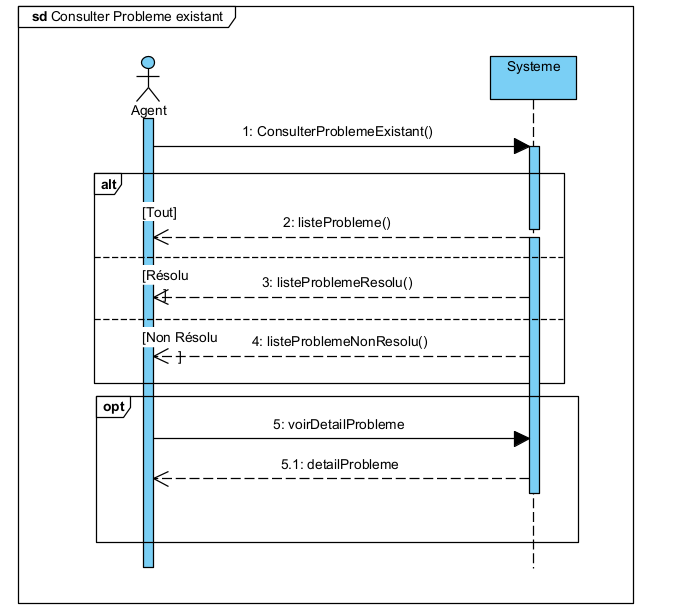
 La Figure 21 est la représentation du diagramme de séquence au niveau du cas d’utiliser « Consulter liste problème existant »

Figure : Diagramme de "séquence "Consulter liste Problème existant "

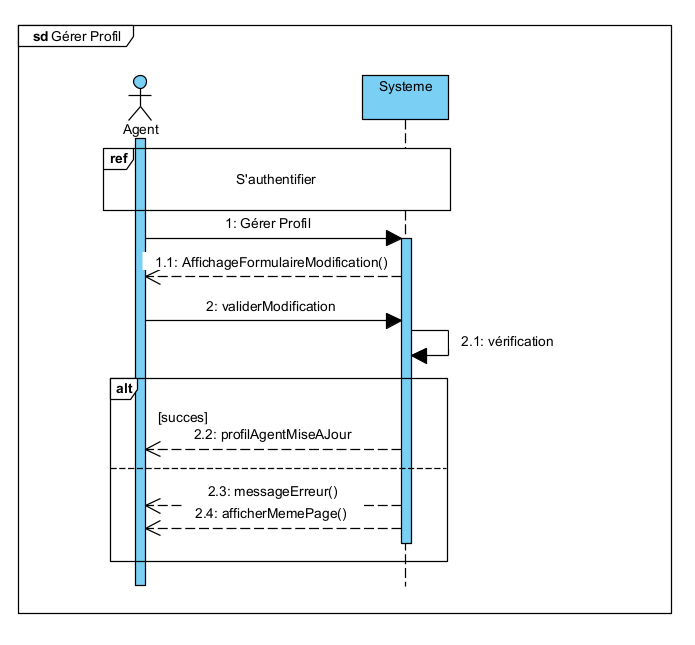
La Figure 22 représente le digramme de séquence du cas d’utilisation « Gérer Profil » .

Figure : Diagramme de séquence "Gérer Profil"

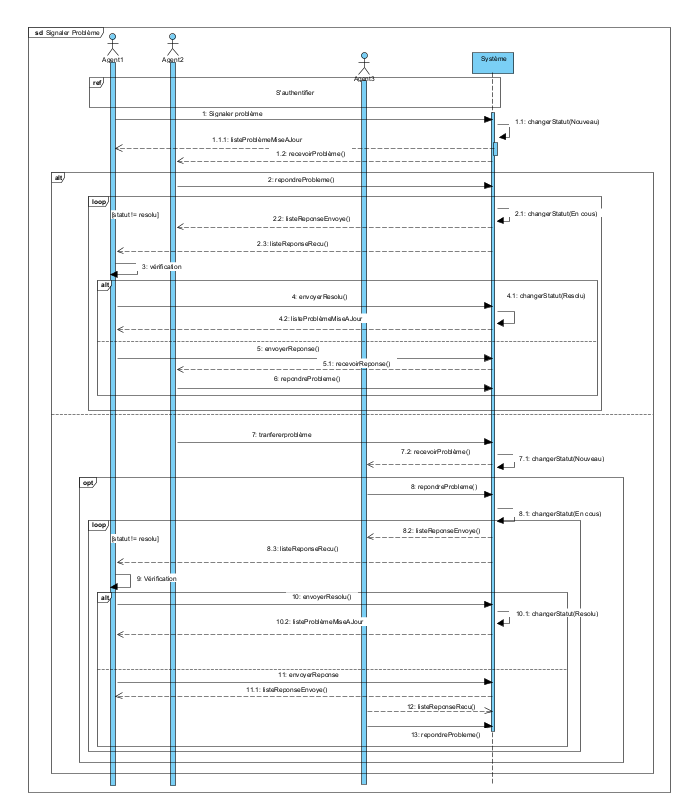
La Figure 23 représente le diagramme de séquence au niveau des cas d’utilisations : « Signaler problème, envoyer réponse, et transférer un problème ».

Figure : Diagramme de séquence "Signaler Problème, Envoyer Réponse, Transférer Problème"



#### **Modèle du domaine**

Le diagramme de classe sert à visualiser la structure du système. Il permet de montrer les principaux composants, sa relation les uns avec les autres et leurs attributs respectifs. Figure 24 représente le diagramme de classe du système

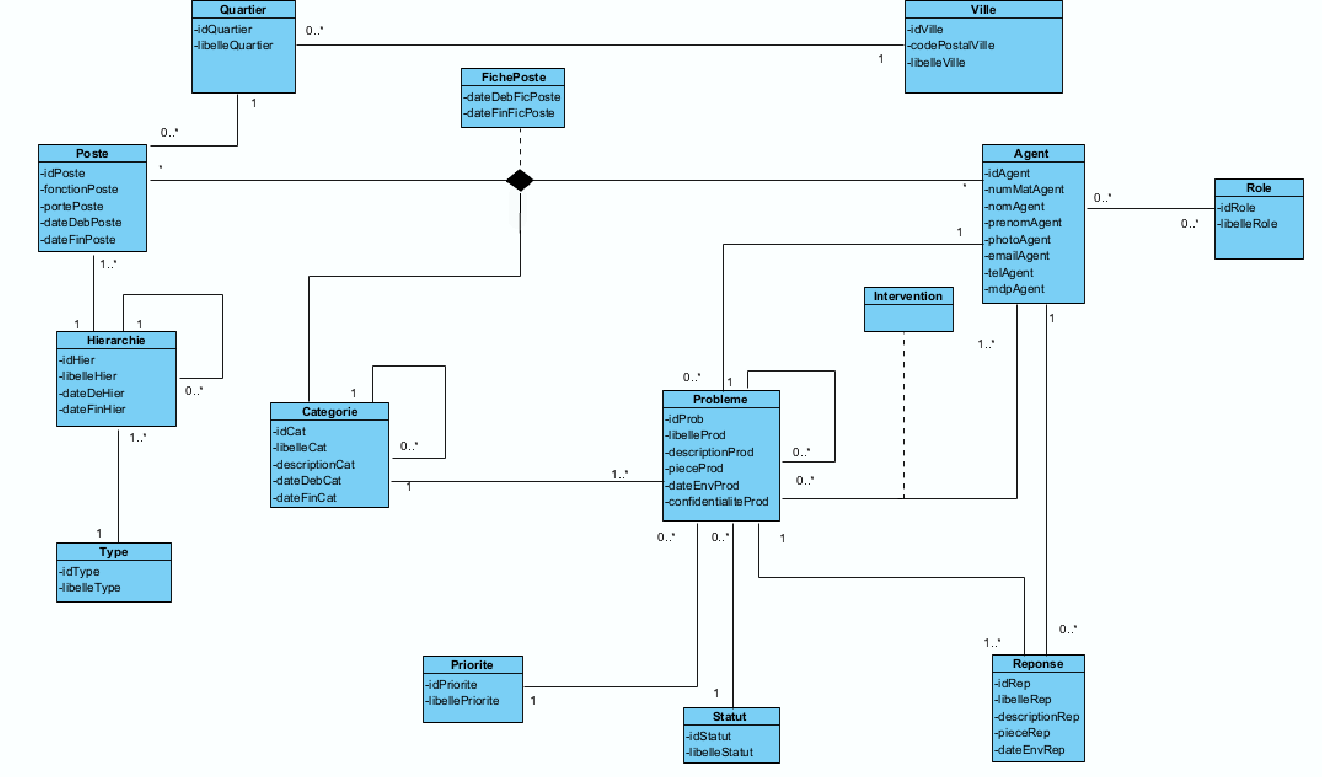


Figure : Diagramme de Classe

## 

## Chapitre 6 : Conception détaillée



### Architecture du système

Une architecture est une structure inhérente à un système informatique, c’est une organisation des différents éléments du système et les relations entre les éléments. Dans ce projet l’architecture prise en compte est l’architecture plus utilisé par la programmation orientée objet qui est « MVC », signifie Modèle Vue Contrôleur, et met en lien l’interface et les modèles grâce au contrôleur. La Figure 27 représente l’architecture de l’application.

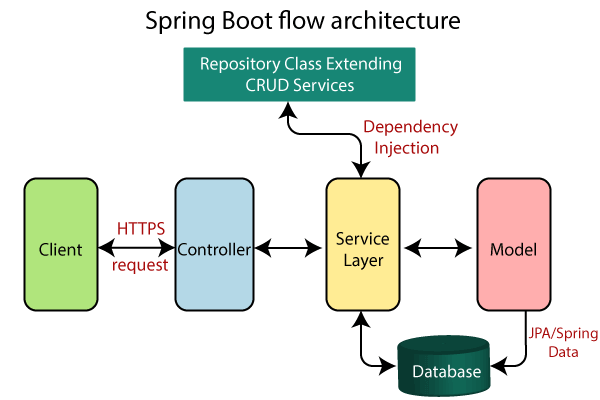


Figure : Architecture du système

### Diagramme de séquence de conception pour chaque cas d’utilisation

La Figure 28 représente le diagramme de séquence de conception du cas d’utilisation « S’authentifier ».

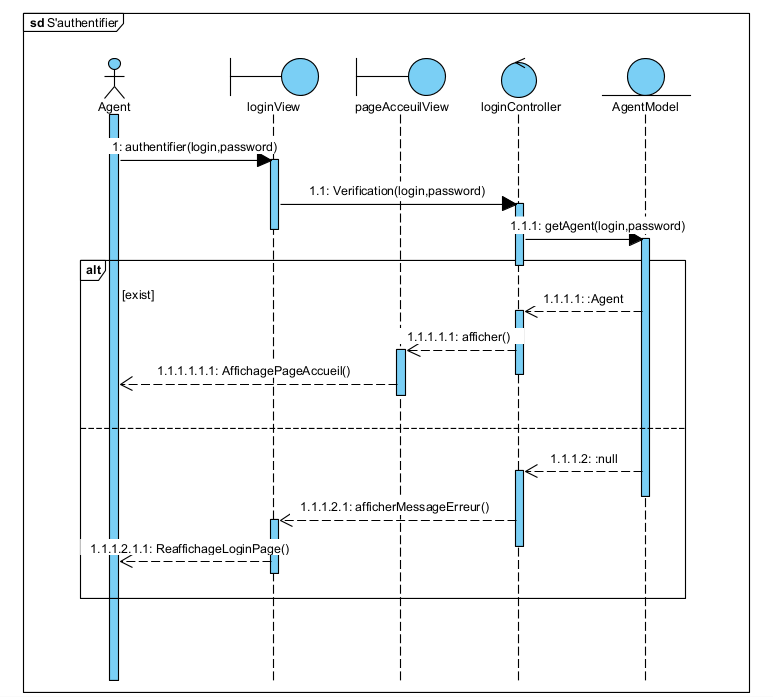


Figure : Diagramme de séquence de Conception "S'authentifier"

La Figure 29 représente le diagramme de séquence de conception du cas d’utilisation « Gérer Agent ».

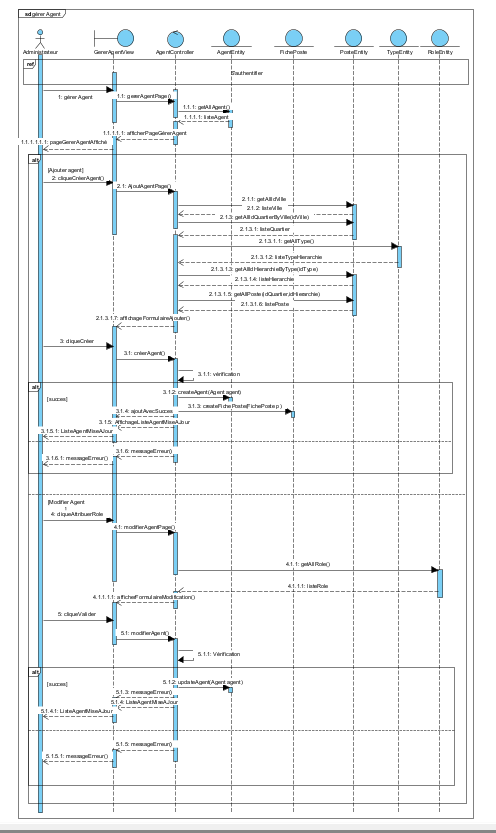


Figure : Diagramme de séquence de conception "Gérer Agent"

Le diagramme de séquence de conception du cas d’utilisation « Gérer Poste » est représenté par la Figure 30 .

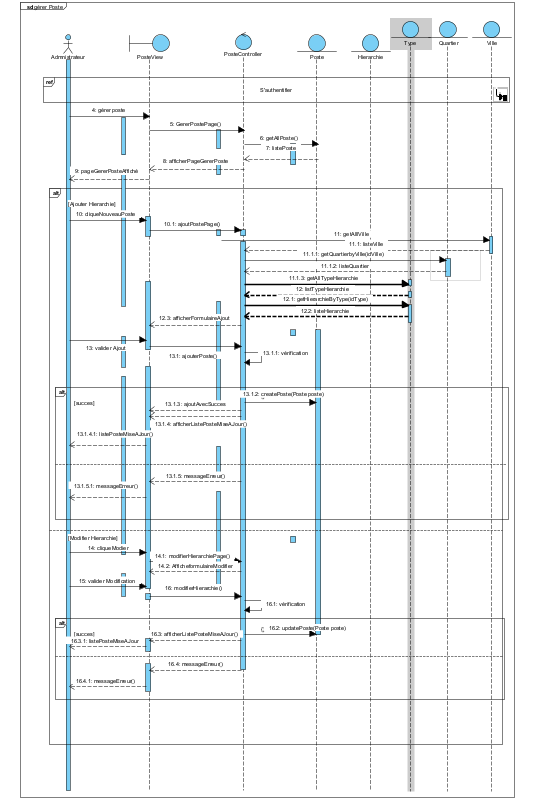


Figure :Diagramme de séquence de conception "Gérer Poste"

La Figure 31 représente le diagramme de séquence de conception du cas d’utilisation « Gérer Hiérarchie ».

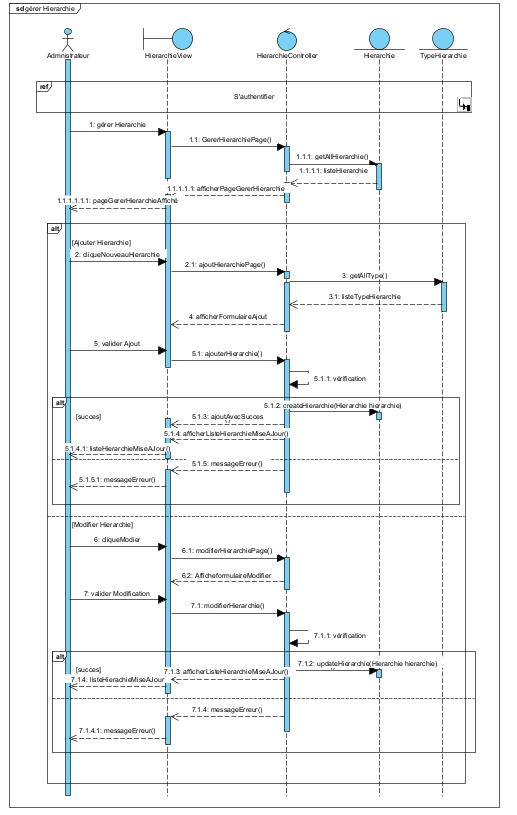


Figure : Diagramme de séquence de conception "Gérer Hiérarchie"

La Figure 32 représente le diagramme de séquence de conception du cas d’utilisation « Gérer Catégorie ».

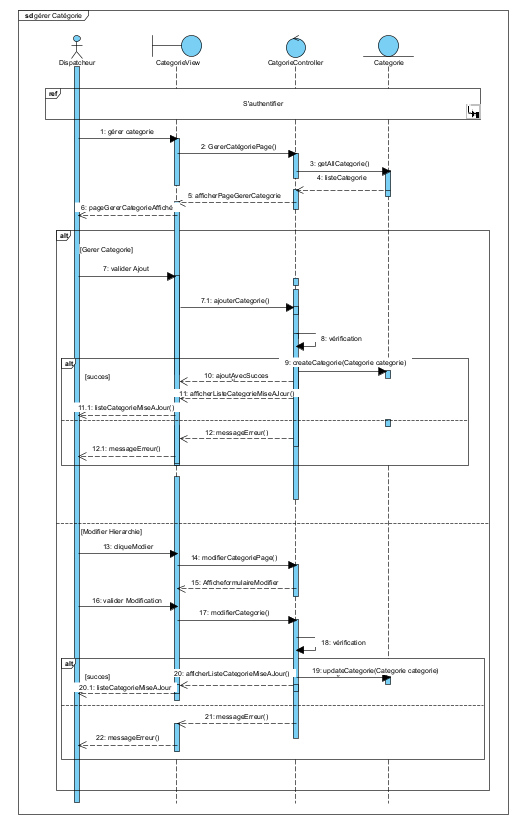


Figure : Diagramme de séquence de conception "gérer Catégorie"

La Figure 33 représente le diagramme de séquence de conception du cas d’utilisation « Gérer Fiche de poste ».

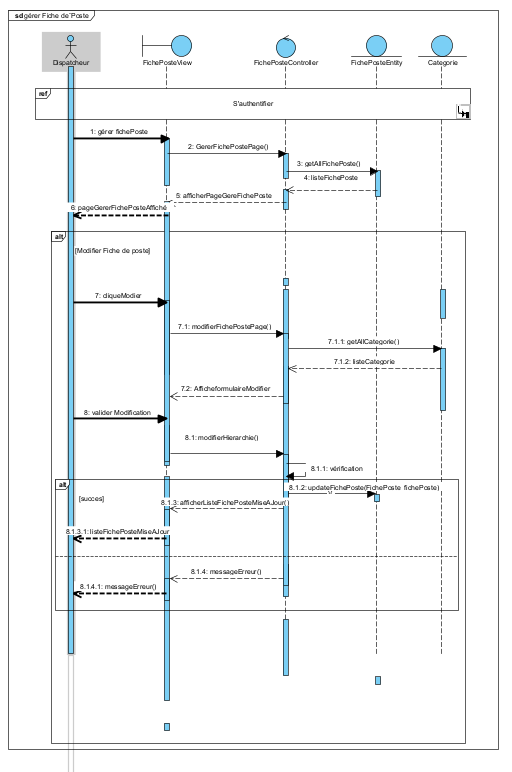


Figure : Diagramme de séquence de conception "Gérer Fiche de Poste"

Le diagramme de séquence de conception du cas d’utilisation « Gérer Profil » est représenté par la Figure 34 .

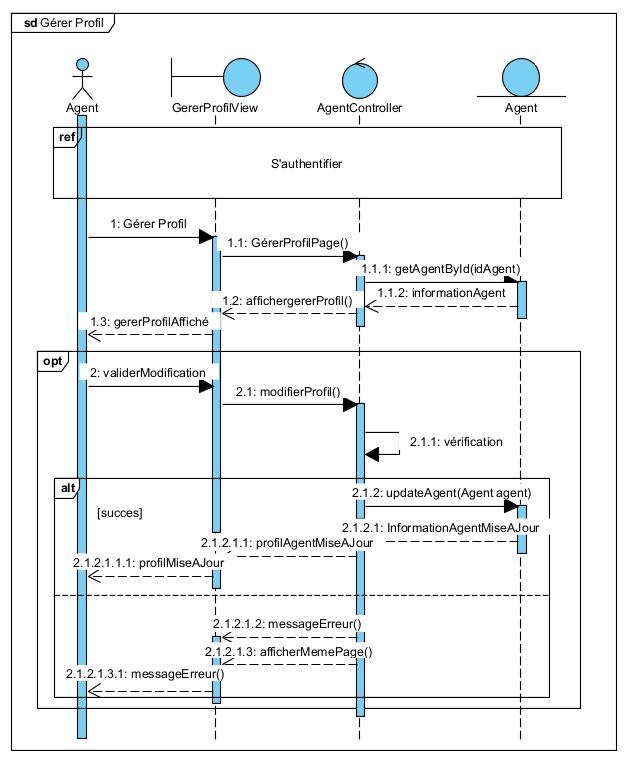


Figure : Diagramme de séquence de conception "Gérer Profil"

Le diagramme de séquence de conception du cas d’utilisation « Consulter liste catégorie et sous-catégorie » est représenté par la Figure 35 .

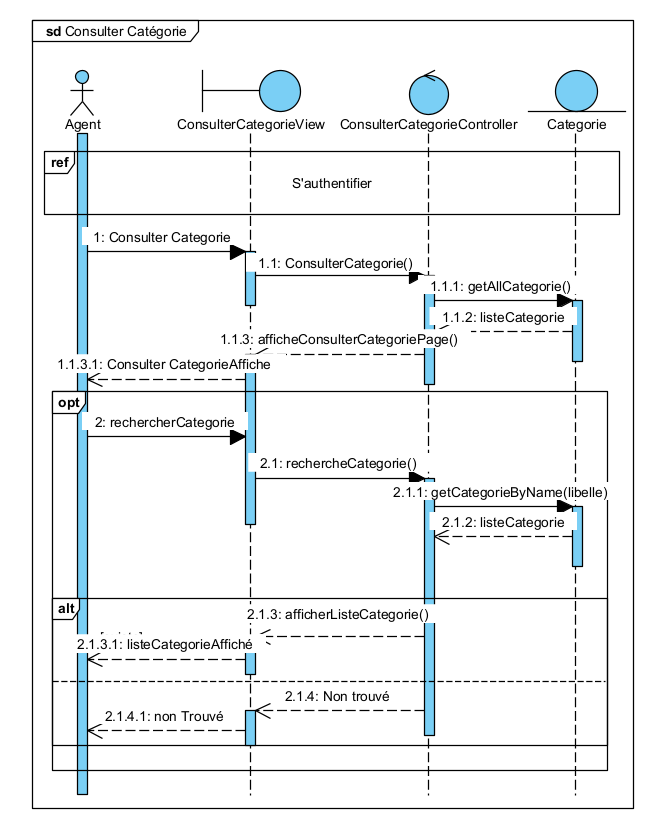


Figure : Diagramme de séquence de conception "Consulter catégorie"

Le diagramme de séquence de conception du cas d’utilisation « Consulter liste problème existant » est représenté par la Figure 36 .

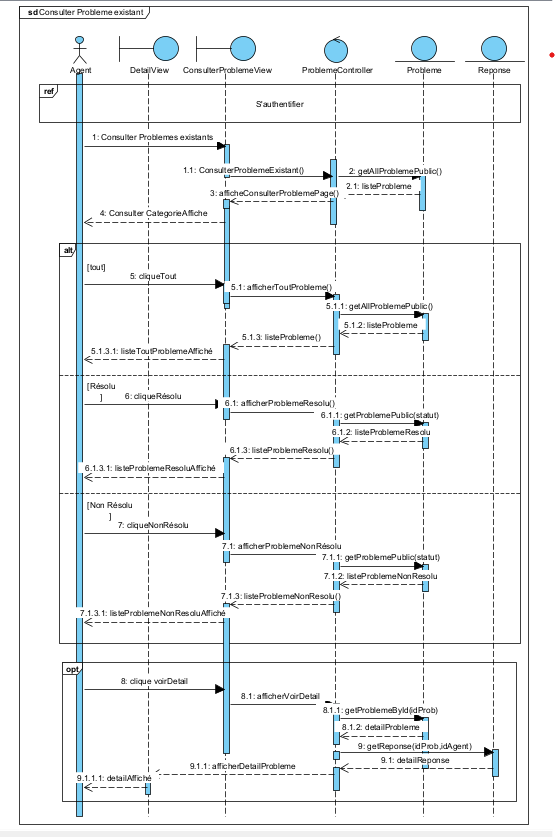


Figure : Diagramme de séquence de conception "Consulter Problème existants"

La Figure 37 montre la représentation du diagramme de séquence de conception au niveau du cas d’utilisation « Signaler problème ».

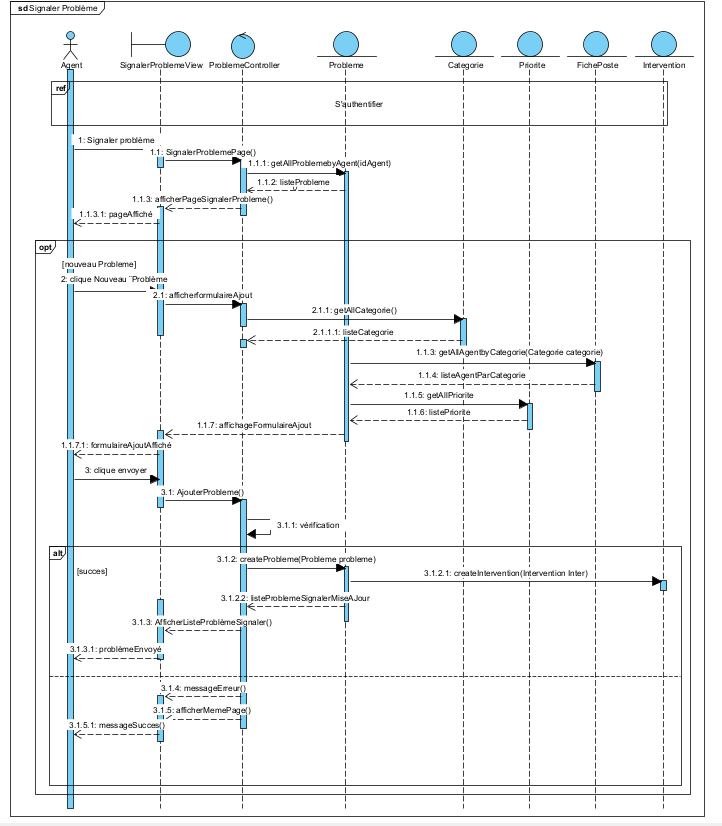


Figure ; Diagramme de séquence de conception "Signaler Problème"

La Figure 38 représente le diagramme de séquence de conception du cas d’utilisation « Envoyer réponse ».

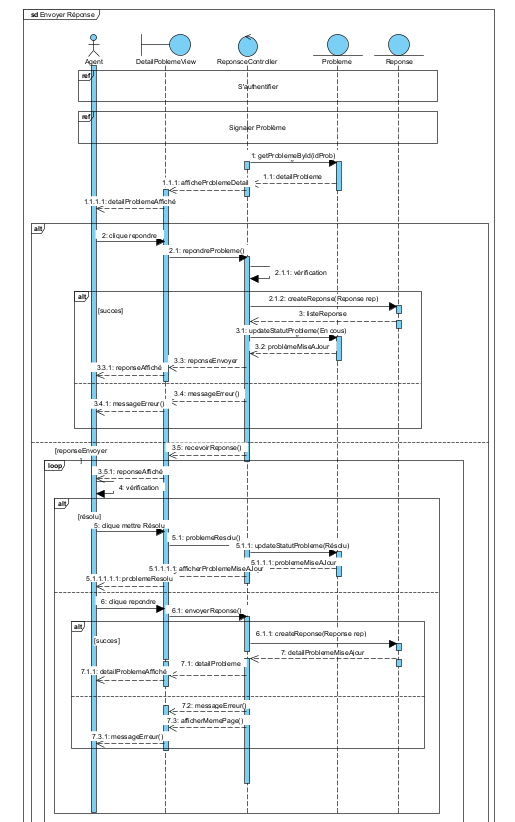


Figure : Diagramme de séquence de conception "Envoyer Réponse"

La Figure 39 représente le diagramme de séquence de conception du cas d’utilisation « Transférer Problème ».

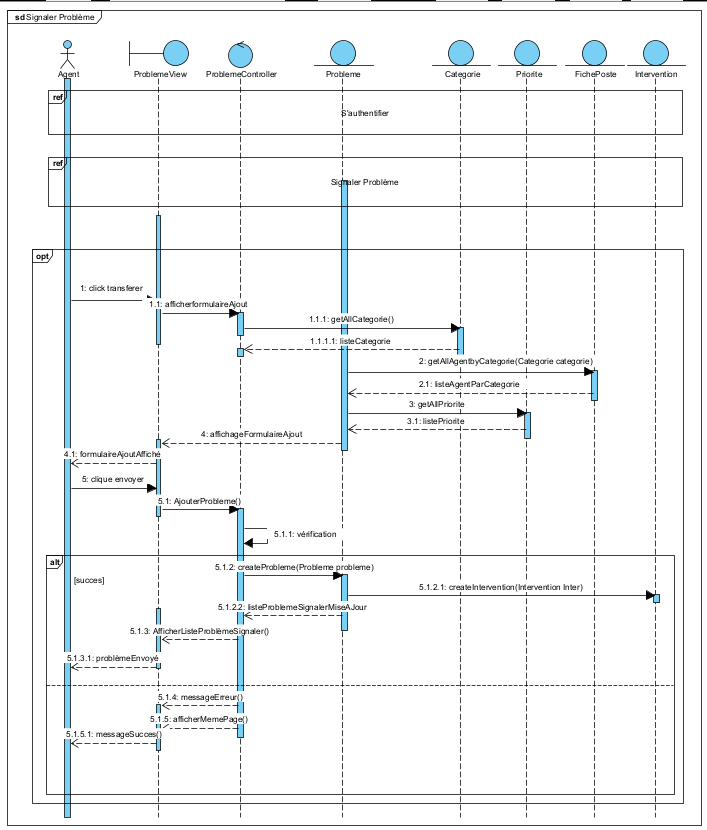


Figure : Diagramme de séquence de conception "Transférer Problème"

### Diagramme de classe de conception pour chaque cas d’utilisation

La Figure 40 montre le diagramme de classe de conception du cas d’utilisation « S’Authentifier ».

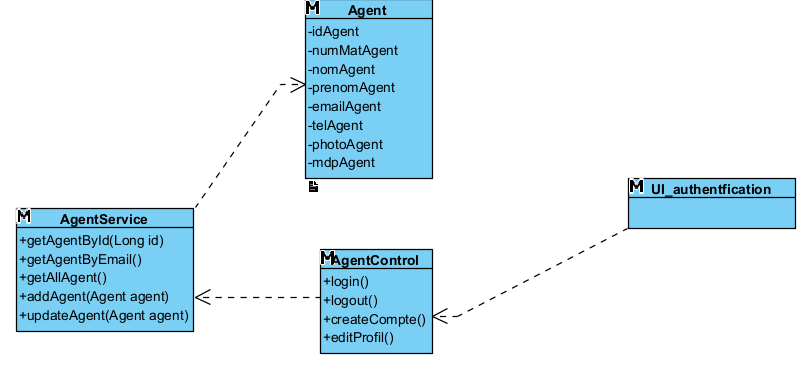


Figure : Diagramme de classe de conception "S'authentifier"

La Figure 41 représente le diagramme de classe de conception du cas d’utilisation « Gérer Hiérarchie ».

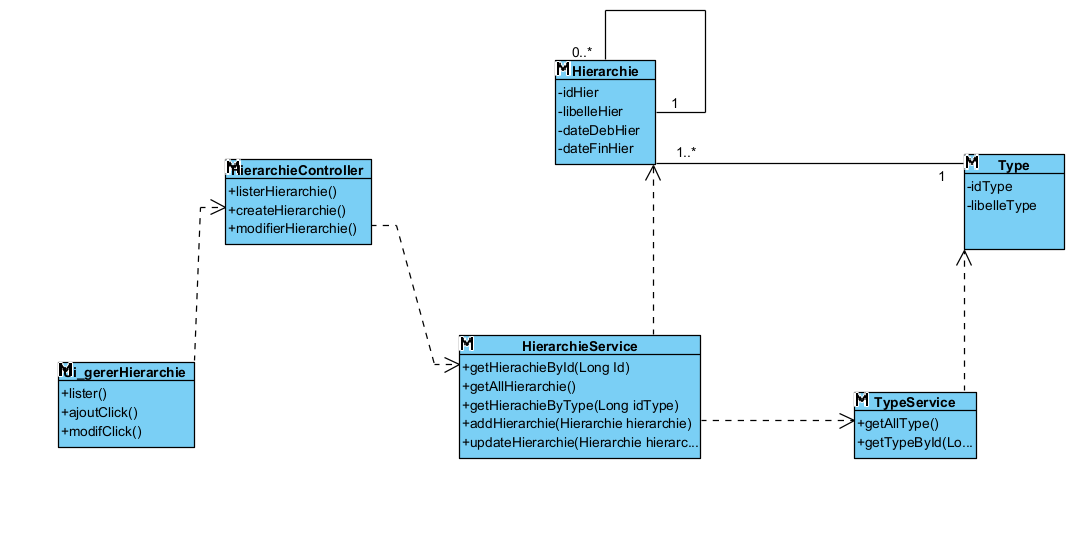


Figure : Diagramme de classe de conception "Gérer Hiérarchie"

La Figure 42 montre le diagramme de classe de conception du cas d’utilisation « Gérer Poste ».

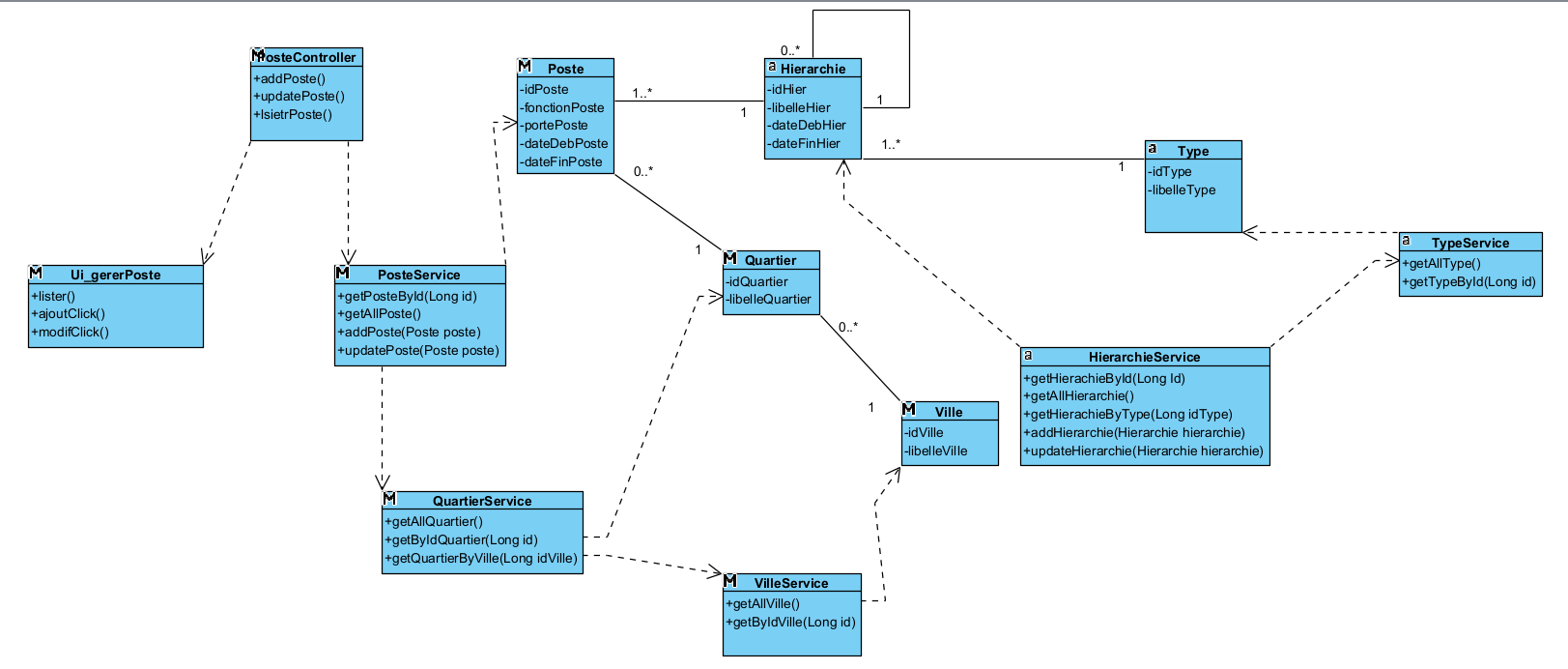


Figure : Diagramme de classe conception "Gérer Poste"

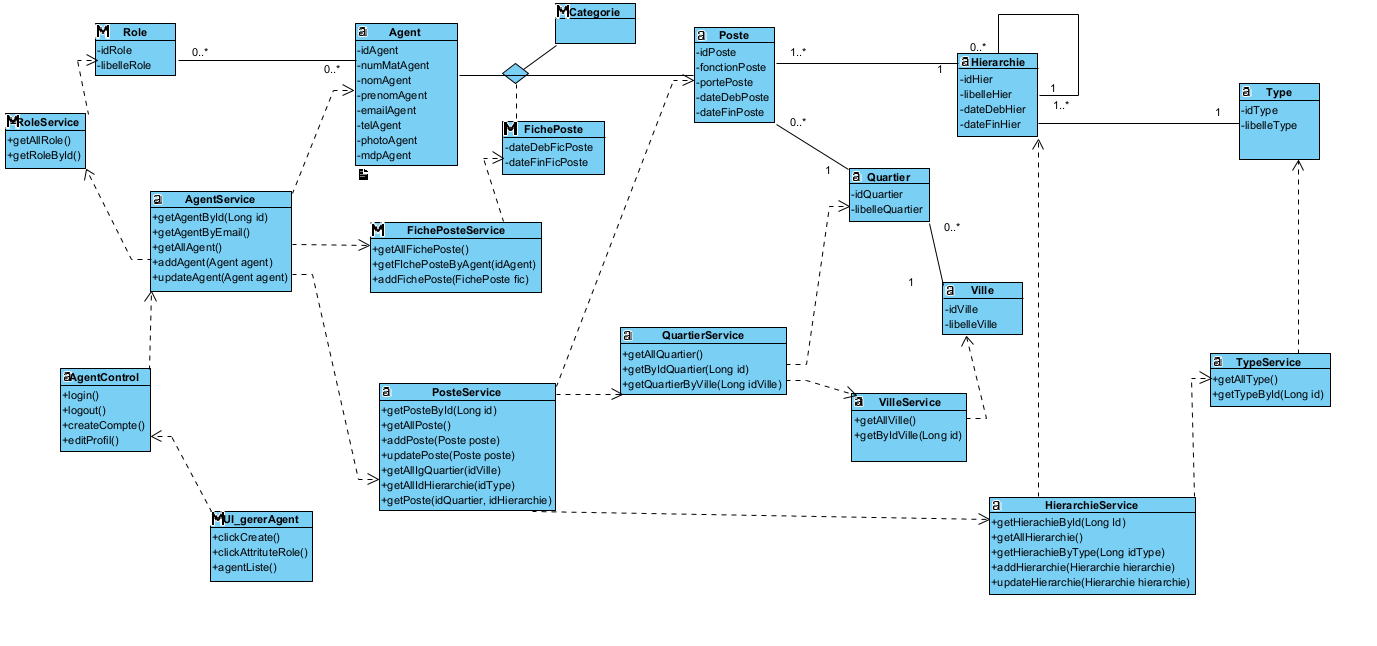
La Figure 41 montre le diagramme de classe de conception du cas d’utilisation « Gérer Agent ».

Figure : Diagramme de classe de conception "Gérer Agent"

La Figure 44 montre le diagramme de classe de conception du cas d’utilisation « Gérer Profil ».

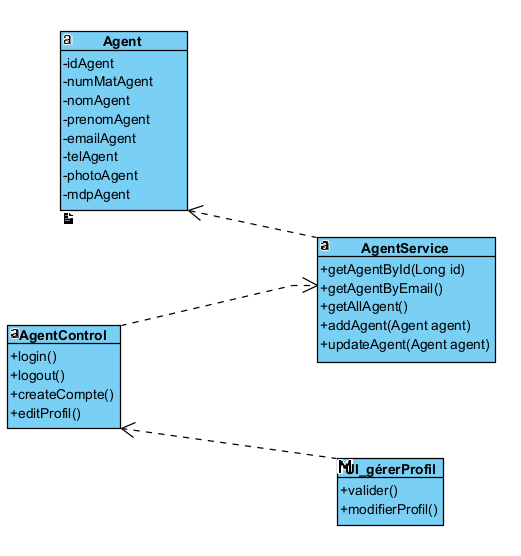


Figure : Diagramme de classe conception "Gérer Profil"

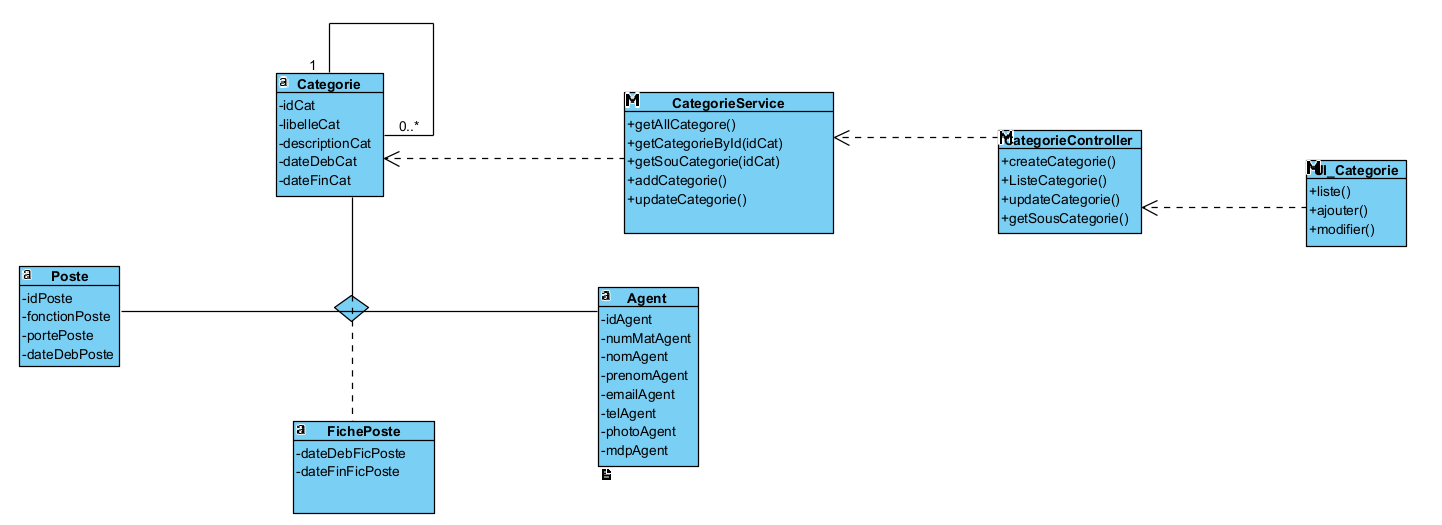
La Figure 45 montre le diagramme de classe de conception du cas d’utilisation « Gérer Catégorie ».

Figure : Diagramme de classe de conception "Gérer Catégorie"

La Figure 46 représente le diagramme de classe de conception au niveau du cas d’utilisation « Gérer Fiche de poste »

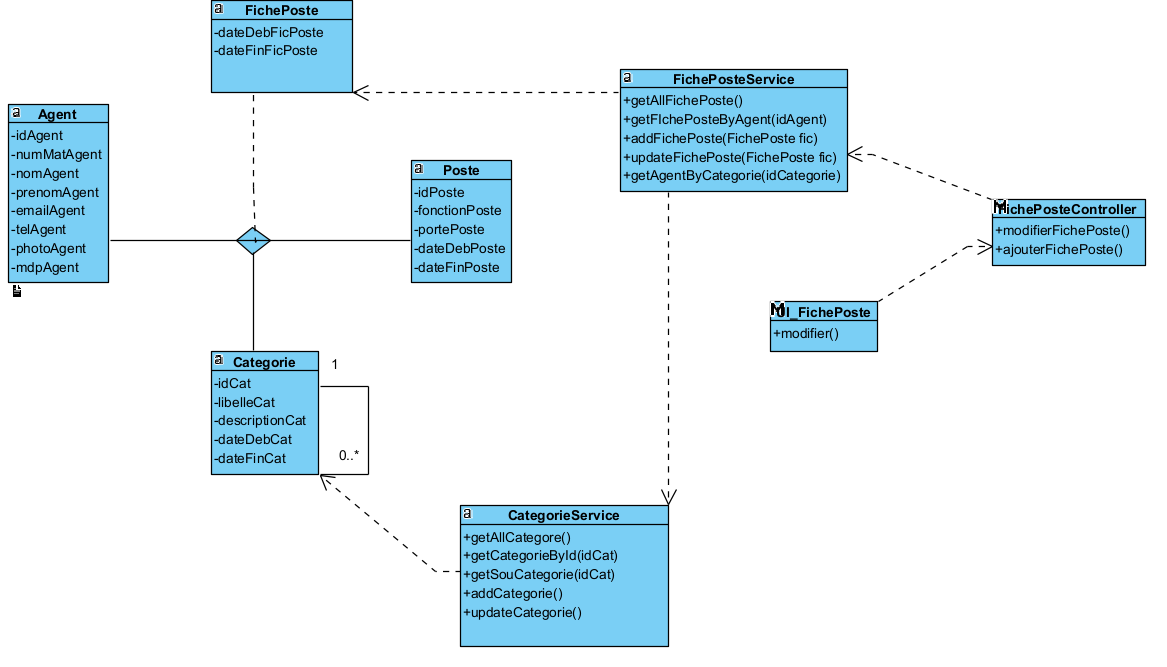


Figure : Diagramme de classe de conception "Gérer Fiche de Poste"

Le diagramme de classe de conception au niveau du cas d’utilisation « Signaler Problème » est représenté par la Figure 47.

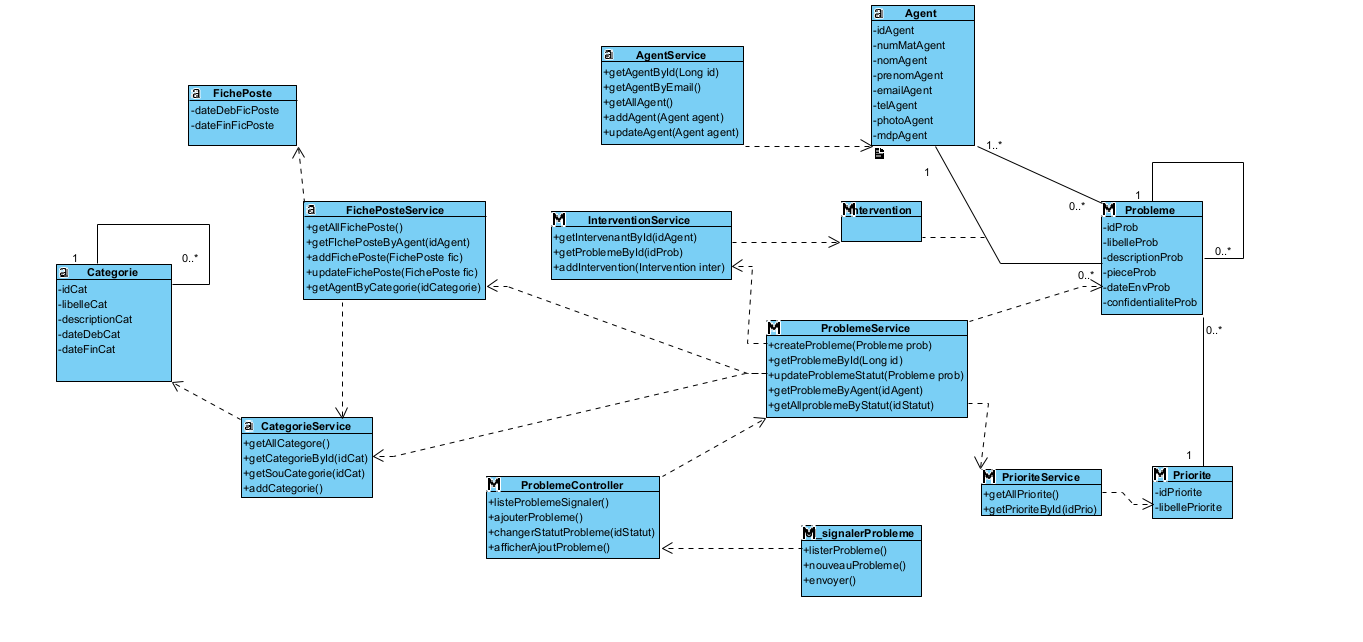
******

Figure : Diagramme de classe de conception "Signaler Problème"

Le diagramme de classe de conception au niveau du cas d’utilisation « Envoyer Réponse » est représenté par la Figure 48.

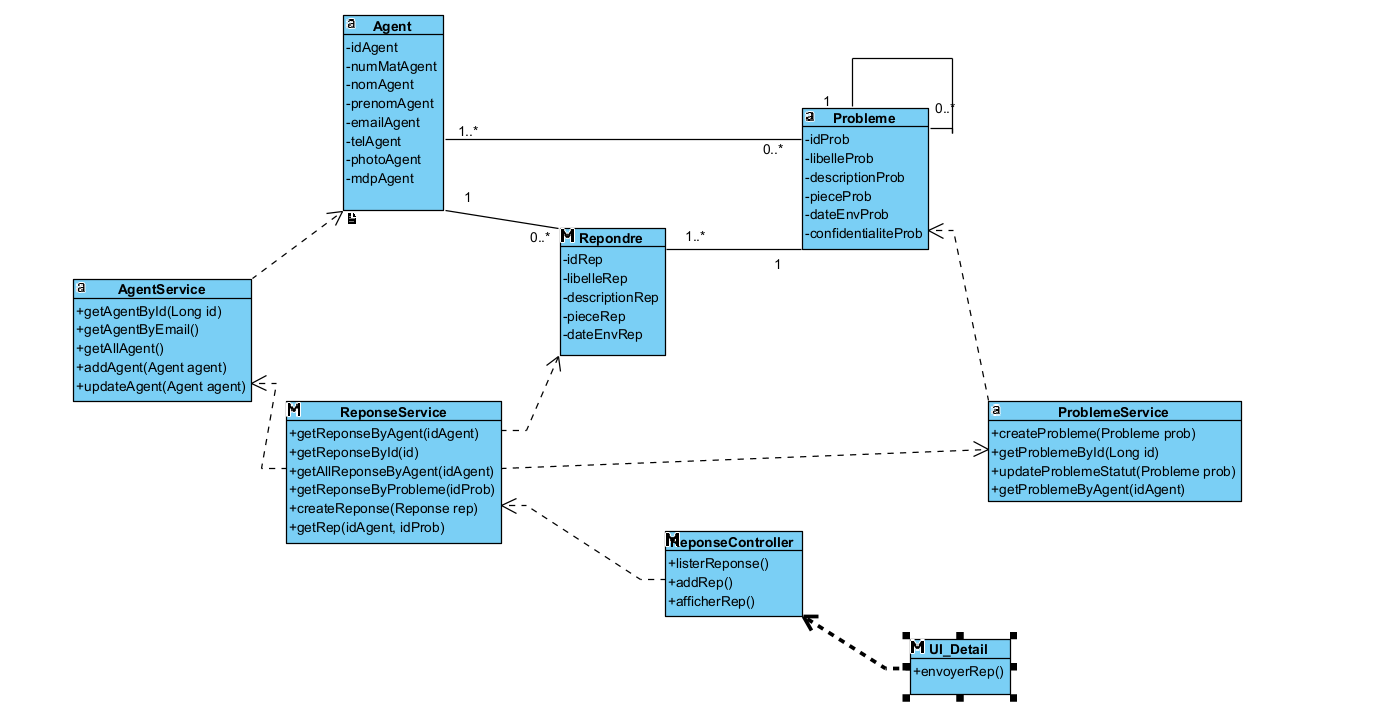
******

Figure : Diagramme de classe de conception "Envoyer réponse"

### Diagramme de classe de conception global

La Figure 49 montre la représentation du diagramme de classe de conception global du système.

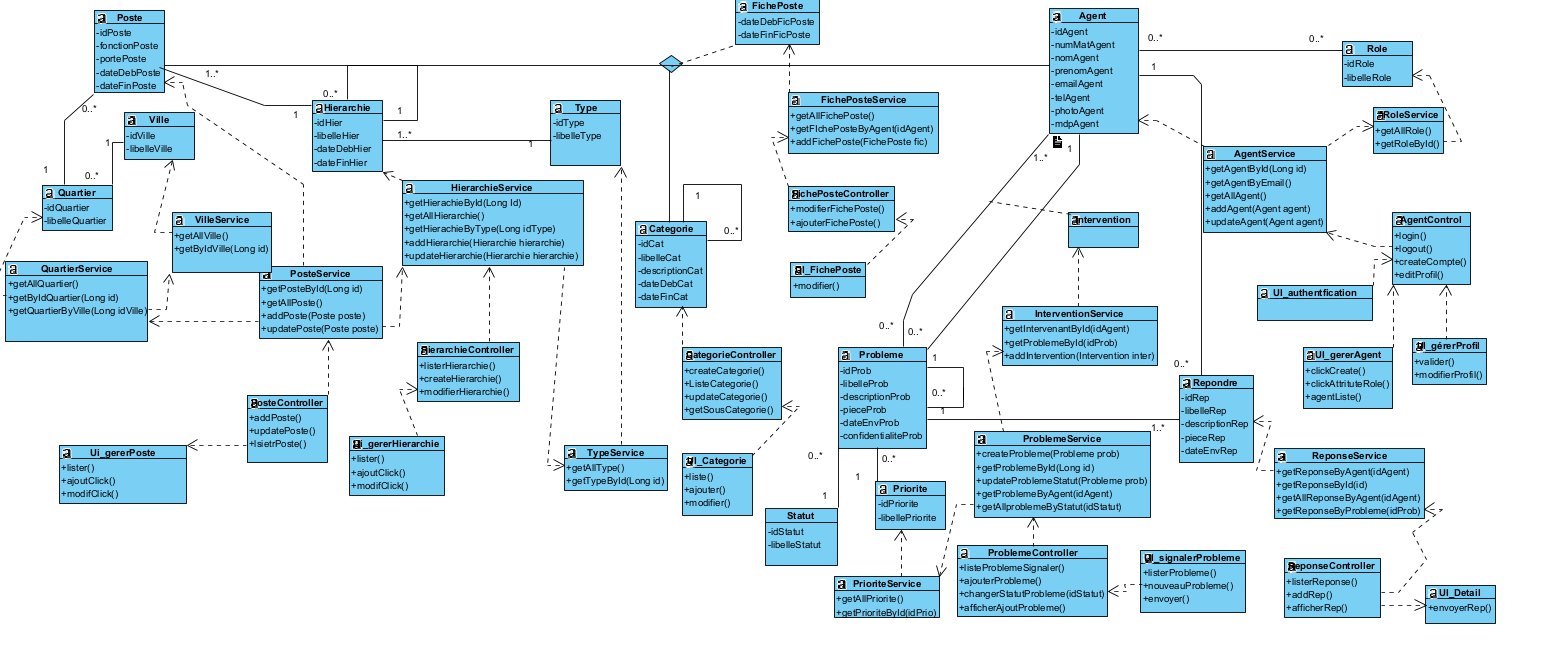


Figure : Diagramme de classe de conception global

### Diagramme de paquetages

Un diagramme de paquetages représente les bibliothèques en relation qui regroupent les classes. Il est la représentation graphique des relations existant entre les paquetages composant un système.

La Figure 50 représente le diagramme de paquetages du système

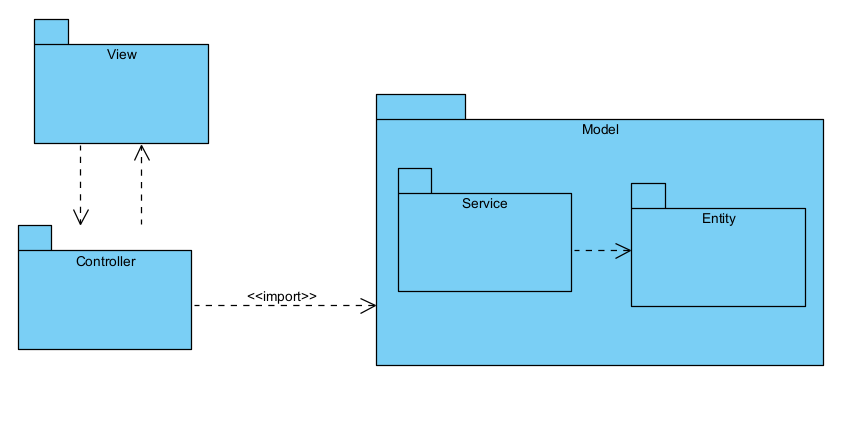


Figure : Diagramme de paquetages

### Diagramme de déploiement

Le diagramme de déploiement représente la modélisation de l’architecture physique d’un système, il affiche d’une part, les relations entre composant logiciels et matériels du système et d’autre part la distribution physique du traitement. La Figure 51 représente le diagramme de déploiement du système.

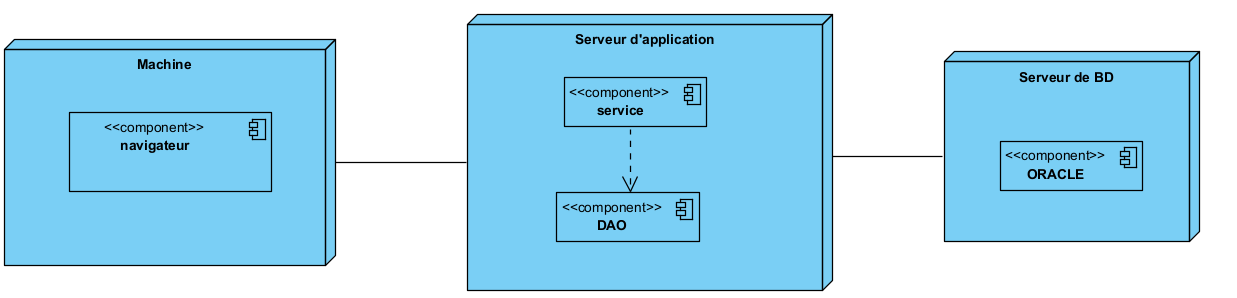


Figure : Diagramme de déploiement

# PARTIE III : REALISATION

## Chapitre 7 : Mise en place de l’environnement de développement

Afin de mettre en place l’environnement de développement, il faut installer et configurer les outils ; il faut, ensuite, adopter une architecture pour l’application.



### Installation et configuration des outils

Les outils utilisés pour la réalisation du projet sont ; Oracle, Visual Paradigm , VS Code.



#### Oracle

Ici la version oracle 11g release 2 sera utilisé comme SGBD. Pour son installation il suffit juste d’un oracle de cette version qui est téléchargeable dans son site officiel en utilisant un compte oracle. La suite consiste d’extraire le fichier et passer aux étapes d’installation qui propose la mise en place du mot de passe administrateur et quelques étapes à suivre jusqu’à ce que l’installation soit terminée. Ces quelques figures ci-après montre l’extrait de l’installation.

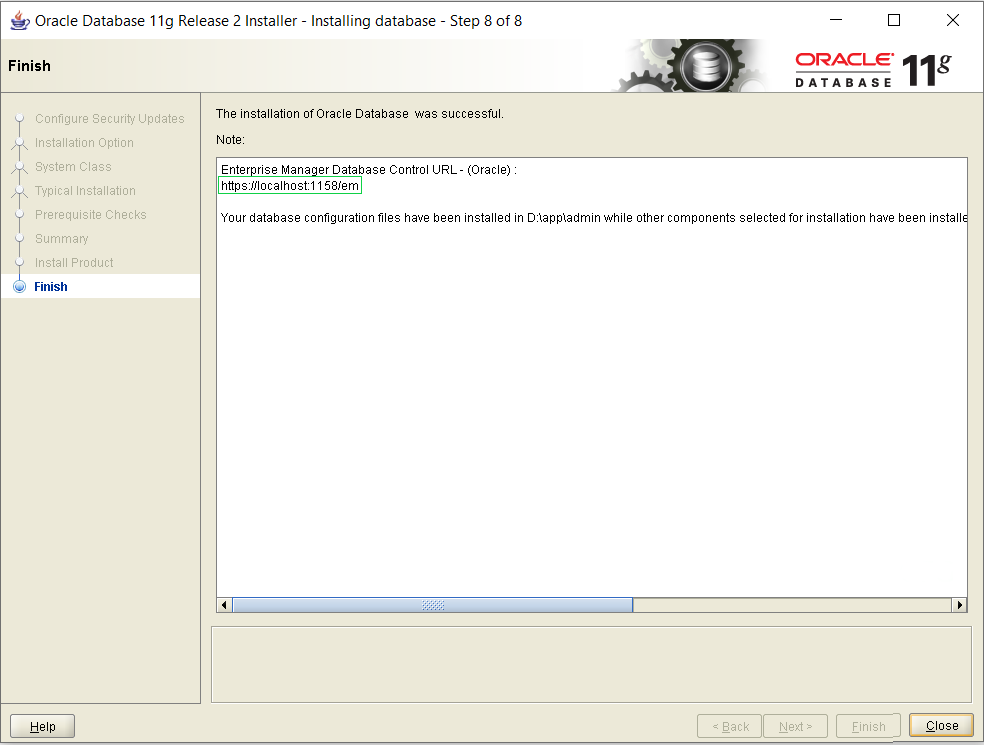
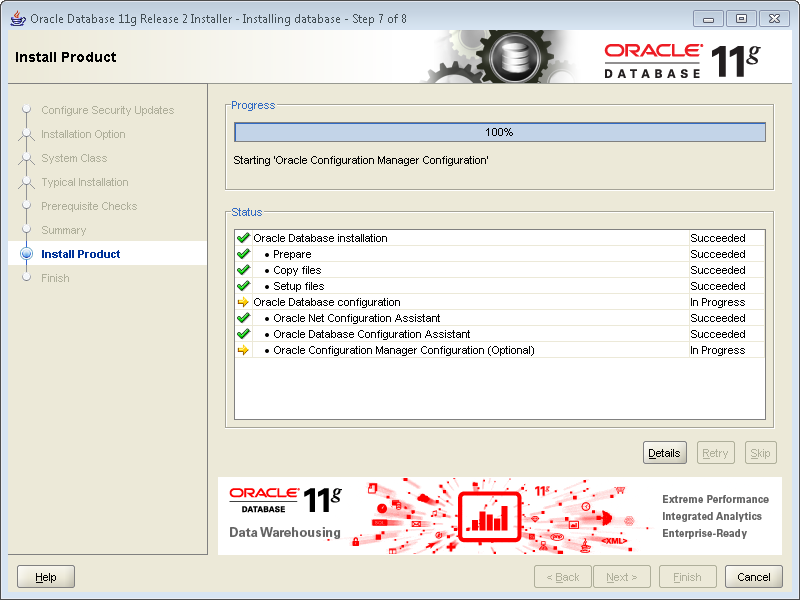
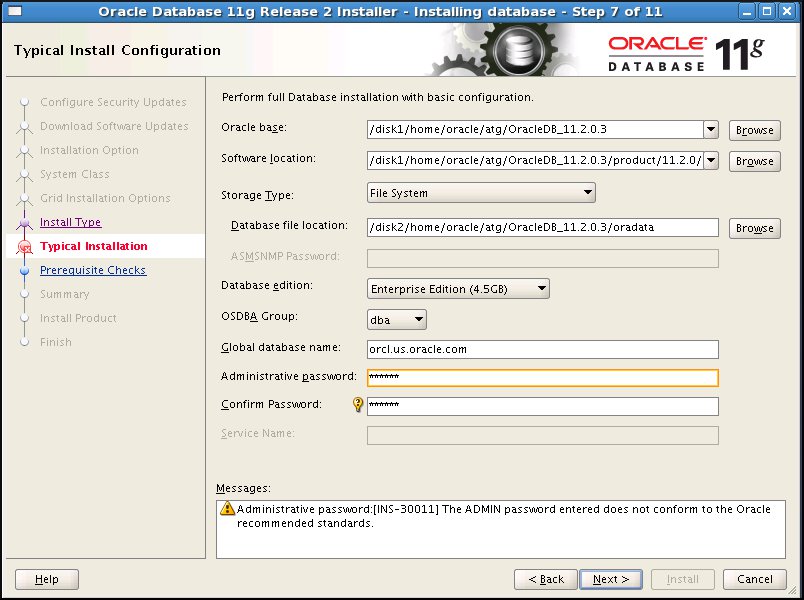
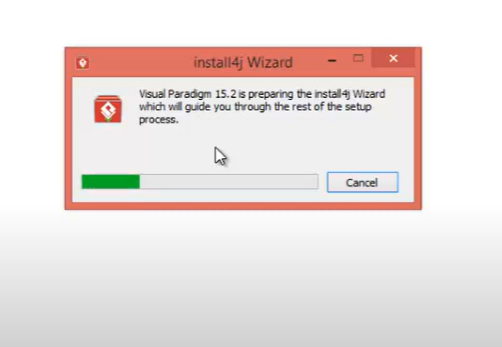
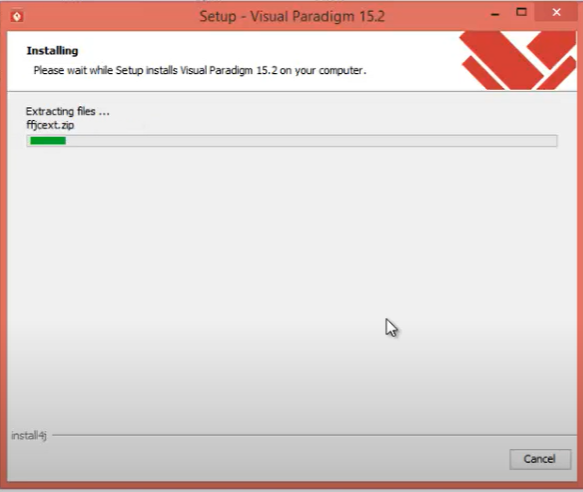
 

Figure : Installation Oracle

#### Visual Paradigm

Visual Paradigm, connu comme outil de modélisation est utilisé dans ce projet comme outil de conception, son installation est si facile vue qu’il est multiplateforme et dispose une version gratuite. La Figure 51 montre quelques étapes de son installation.



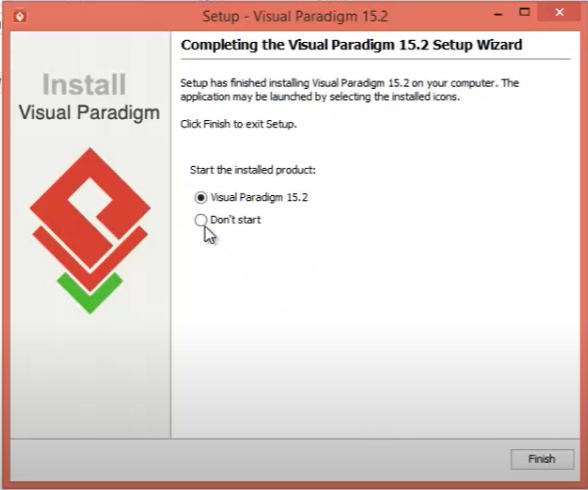
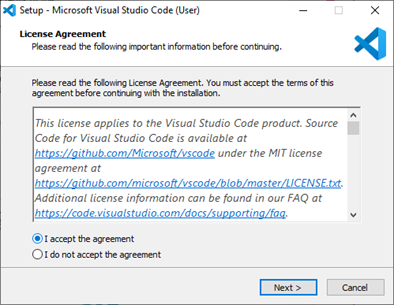


Figure : Installation Visual Paradigm

#### Visual Studio Code

Visual studio sera l’environnement de développement utile pour programmer l’application. Son installation suffit juste de suivre les étapes indiquées. Mais pour intégrer Spring Boot et le langage java, il faudrait installer quelques extensions comme : « Extension Pack Java , Spring Boot Extension Pack » . La Figure 52 montre un extrait de l’installation de cet outil et celui des extensions utiles.

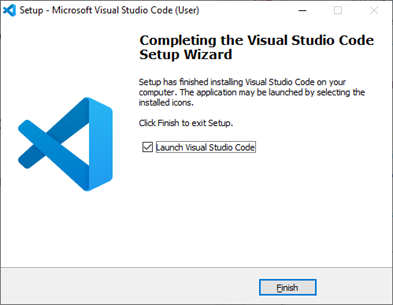
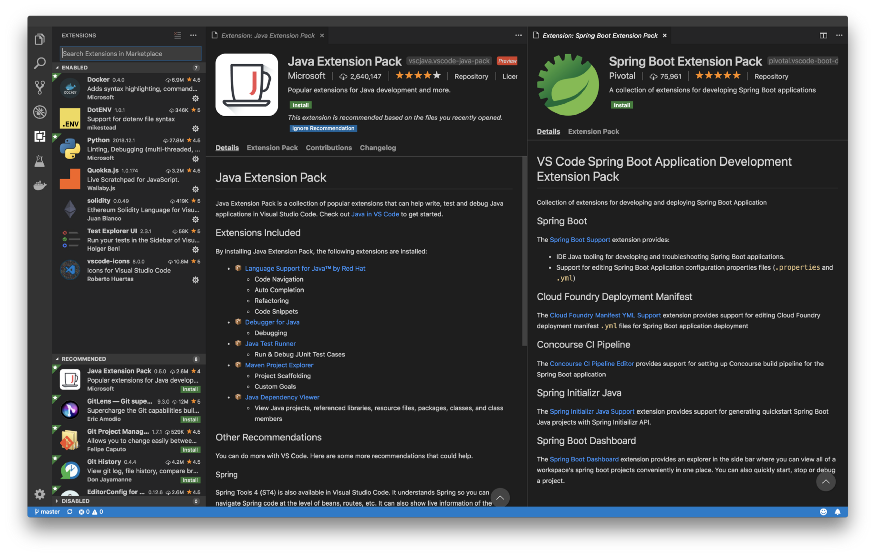


Figure : Installation Visual Studio Code

### Architecture de l’application

L’application utilise l’architecture MVC, c’est-à-dire se sert de 3 couches fondamentales pour son fonctionnement : La couche Model, Vue, et contrôleur.

* Modèle : présente les classes et services utilisés dans cette application,
* Vue : Couche supérieure qui compose la partie frontale de l’application, gère les requête http et effectue l’authentification, elle est responsable de la conversion du paramètre du champs JSON en objets Java ou vice-versa. Ici elle est représentée par un moteur de template écrit en JAVA dit « Thymeleaf »
* Contrôleur : couche qui reçoit la requête http, traite les descripteurs et appelle la logique du serveur

La Figure 55 montre la partie plus détaillée.

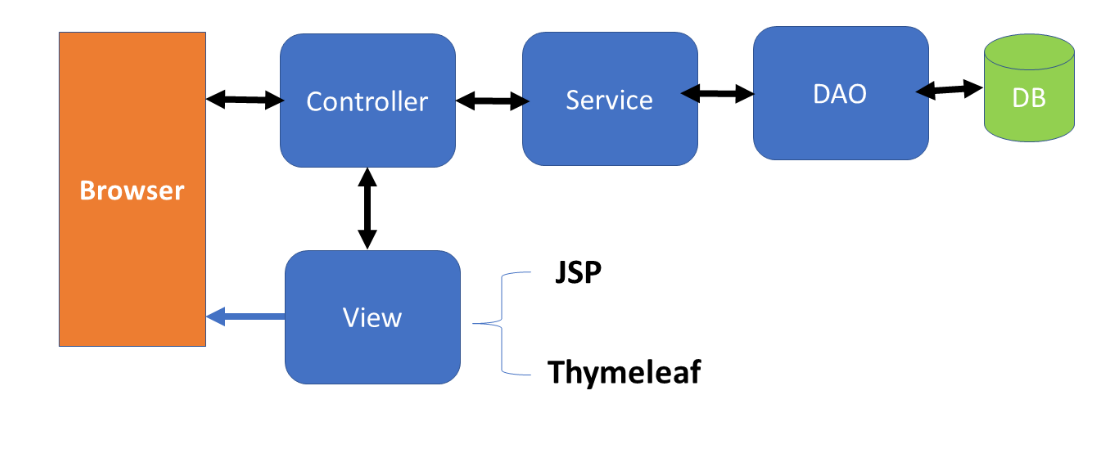


Figure : Architecture de l'application

## Chapitre 8 : Développement de l’application



### Création de la base de données

La création de base de données se fait en se connectant avec SQL DEVELOPPER en tant qu’administrateur et en créant une nouvelle connexion. Puis il suffit de la liée dans le code spring boot en écrivant quelques lignes de code dans le fichier application propreties de spring boot c’est ce qui est représenté dans la Figure 56. Après il n’aura plus rien à craindre car JPA prend en charge la création automatique des tables à partir des classes créées pour l’application. La montre la classe qui va être générée automatiquement en table et la Figure 56 montre sa représentation dans la base de données oracle après sa création.

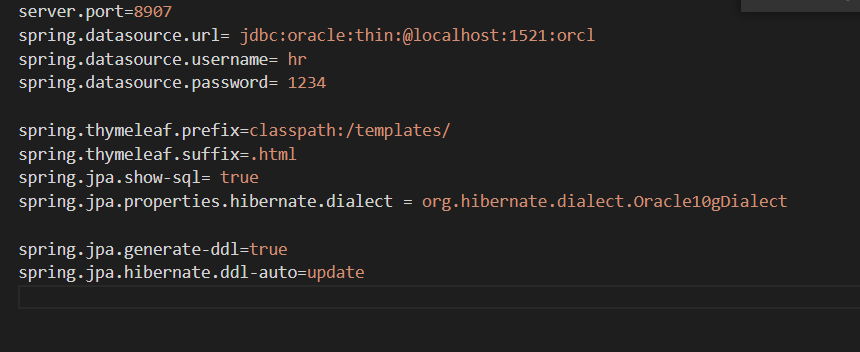


Figure : Connexion Base de Données Oracle

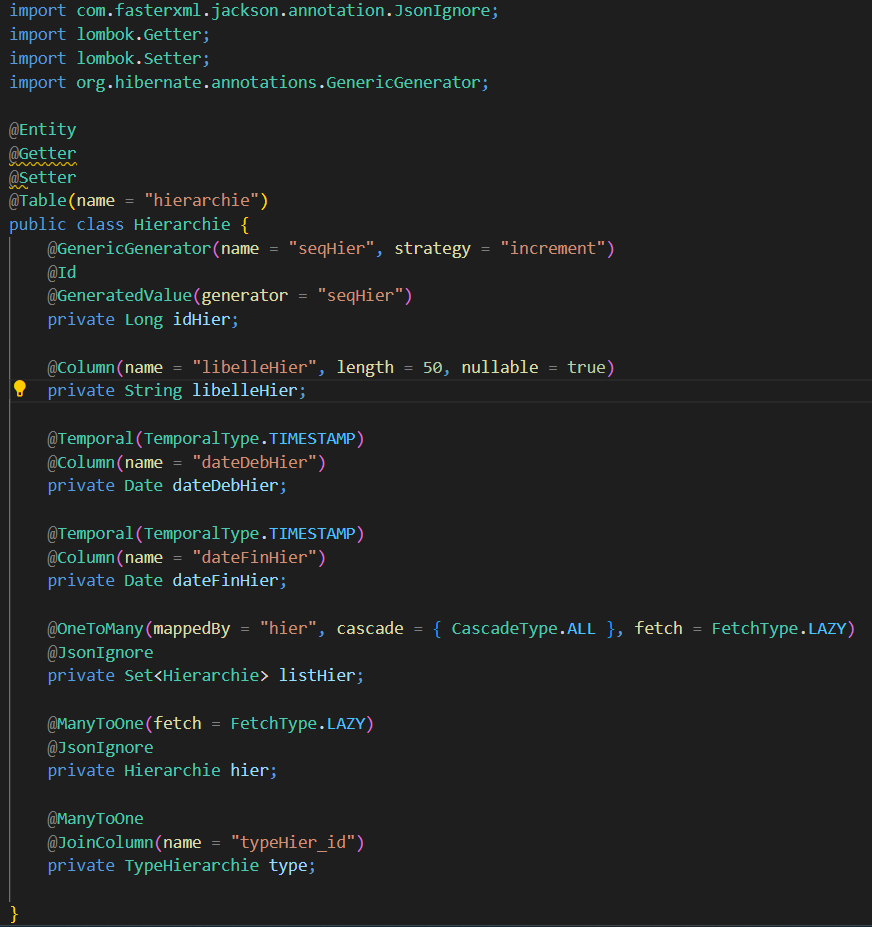


Figure : Classe hiérarchie

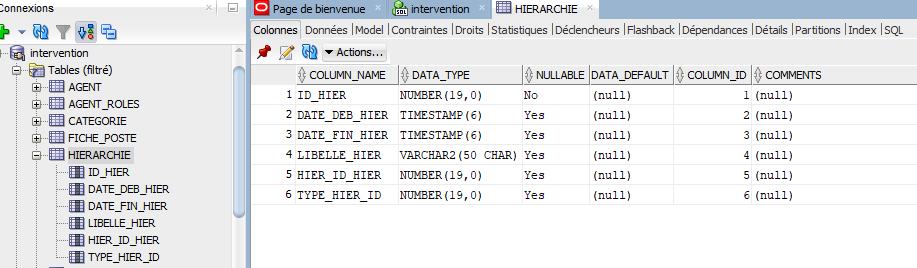
******

Figure : Hiérarchie dans la base de données



### Codage de l’application

Dans cette partie nous allons voir un petit coup d’œil quelques échantillon du code qui structure l’application.



#### Présentation d’un service

* Service Interface Catégorie

package mg.douane.intervention.service;

import mg.douane.intervention.data.domaine.Categorie;

import java.util.List;

public interface CategorieService {

public Iterable<Categorie> getAllCategories() ;

public List<Categorie> getAllSousCategoriesByCat(Long id);

 public Categorie createCategorie(Categorie cat) throws Exception ;

 public Categorie updateCategorie(Categorie cat) throws Exception ;

}

* Service Categorie

package mg.douane.intervention.service;

import mg.douane.intervention.data.domaine.Categorie;

import mg.douane.intervention.repository.CategorieRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class CategorieServiceImpl implements CategorieService {

    @Autowired

    CategorieRepository categoriRepository;

    @Override

    public Iterable<Categorie> getAllCategories() {

        return categoriRepository.findAllCategories();

    }

//getSousCategorie

    @Override

    public List<Categorie> getAllSousCategoriesByCat(Long id) {

        Optional<Categorie> categorie = categoriRepository.findById(id);

        if (categorie.isPresent()) {

            List<Categorie> cat = categoriRepository.findAllByCat(categorie.get());

            return cat;

        }

        return null;

    }

//ajout categorie service

     @Override

    public Categorie createCategorie(Categorie cat) throws Exception {

        cat = CategorieRepository.save(hier);

        return cat;

    }

//update Categorie

  @Override

    public Categorie updateCategorie (Categorie cat) {

        return categorie Repository.save(cat);

    }}

#### Exemple de Controller Agent

package mg.douane.intervention.controller;

import mg.douane.intervention.data.domaine.\*;

import mg.douane.intervention.data.dto.ChangePasswordDto;

import mg.douane.intervention.data.dto.HierarchieDto;

import mg.douane.intervention.data.dto.PosteDto;

import mg.douane.intervention.data.dto.QuartierDto;

import mg.douane.intervention.repository.\*;

import mg.douane.intervention.service.AgentService;

import mg.douane.intervention.service.HierarchieService;

import mg.douane.intervention.service.PosteService;

import mg.douane.intervention.service.QuartierService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.validation.Errors;

import org.springframework.web.bind.annotation.\*;

import javax.validation.Valid;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import java.util.Optional;

import java.util.stream.Collectors;

@Controller

public class AgentController {

    @Autowired

    AgentService agentService;

    @Autowired

    RoleRepository roleRepository;

    @Autowired

    TypeHierarchieRepository typeHierarchieRepository;

    @Autowired

    HierarchieRepository hierarchieRepository;

    @Autowired

    HierarchieService hierarchieService;

    @Autowired

    PosteRepository posteRepository;

    @Autowired

    VilleRepository villeRepository;

    @Autowired

    QuartierService quartierService;

    @Autowired

    CategorieRepository categorieRepository;

    @Autowired

    FichePosteRepository fichePosteRepository;

    @Autowired

    PosteService posteService;

    @RequestMapping(value = { "/", "/login" })

    public String index() {

        return "index";

    }

    // GERER AGENT

    @RequestMapping(value = "/agent")

    public String gereAgentPage(Model model) {

        model.addAttribute("agentList", agentService.getAllAgents());

        model.addAttribute("roleList", roleRepository.findAll());

        model.addAttribute("agentForm", new Agent());

        model.addAttribute("fichePosteForm", new FichePoste());

        model.addAttribute("agentList", agentService.getAllAgents());

        model.addAttribute("typeHierarchie", typeHierarchieRepository.findAll());

        model.addAttribute("postes", posteRepository.findAll());

        model.addAttribute("villes", villeRepository.findAll());

        model.addAttribute("souscateg", categorieRepository.findAll());

        return "Admin/GererAgent";

    }

 @PostMapping("/agentForm")

    public String createAgent(@Valid @ModelAttribute("agentForm") Agent agent,

            @ModelAttribute("fichePosteForm") FichePoste fichePoste, BindingResult result, ModelMap model) {

        if (result.hasErrors()) {

            model.addAttribute("agentForm", agent);

            model.addAttribute("formTab", "active");

        } else {

            try {

                Agent agnt = agentService.createAgent(agent);

                Optional<Poste> poste = posteRepository.findById(fichePoste.getPosteFich().getIdPoste());

                fichePoste.setAgentFich(agnt);

                fichePoste.setPosteFich(poste.get());

                fichePoste.setDateDebFich(new Date()) ;

                fichePosteRepository.save(fichePoste);

                model.addAttribute("agentForm", new Agent());

                model.addAttribute("listTab", "active");

            } catch (Exception e) {

                model.addAttribute("formErrorMessage", e.getMessage());

                model.addAttribute("agentForm", agent);

                model.addAttribute("formTab", "active");

                model.addAttribute("agentList", agentService.getAllAgents());

                model.addAttribute("roles", roleRepository.findAll());

            }

        }

        model.addAttribute("agentList", agentService.getAllAgents());

        model.addAttribute("roles", roleRepository.findAll());

        return "Admin/GererAgent";

    }

### Présentation de l’application

Des captures d’écran présenteront des aperçus de l’interface de l’application réalisée.

#### Page d’authentification

Figure : Page d'Authentification

#### Page gérer Agent

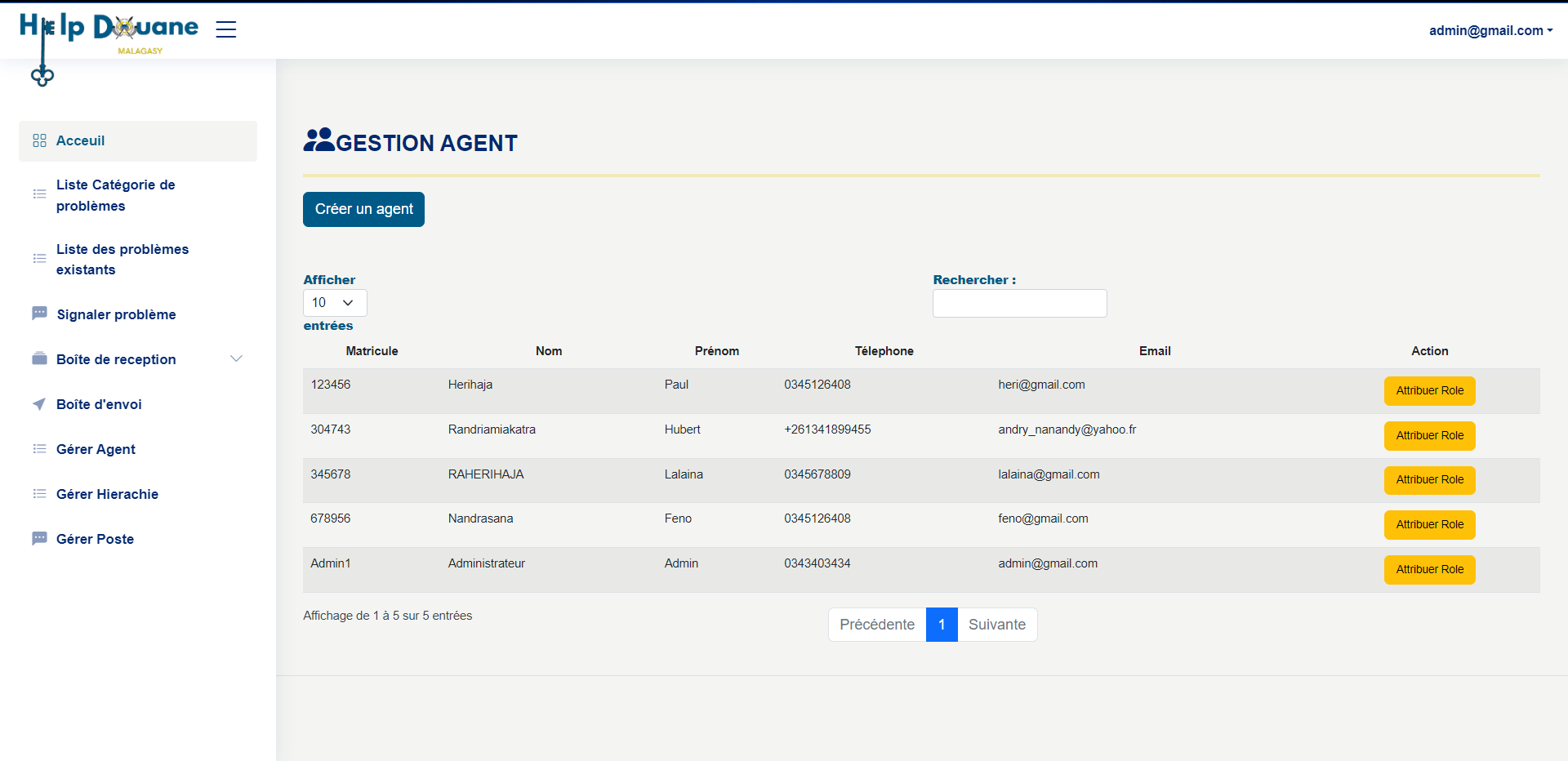
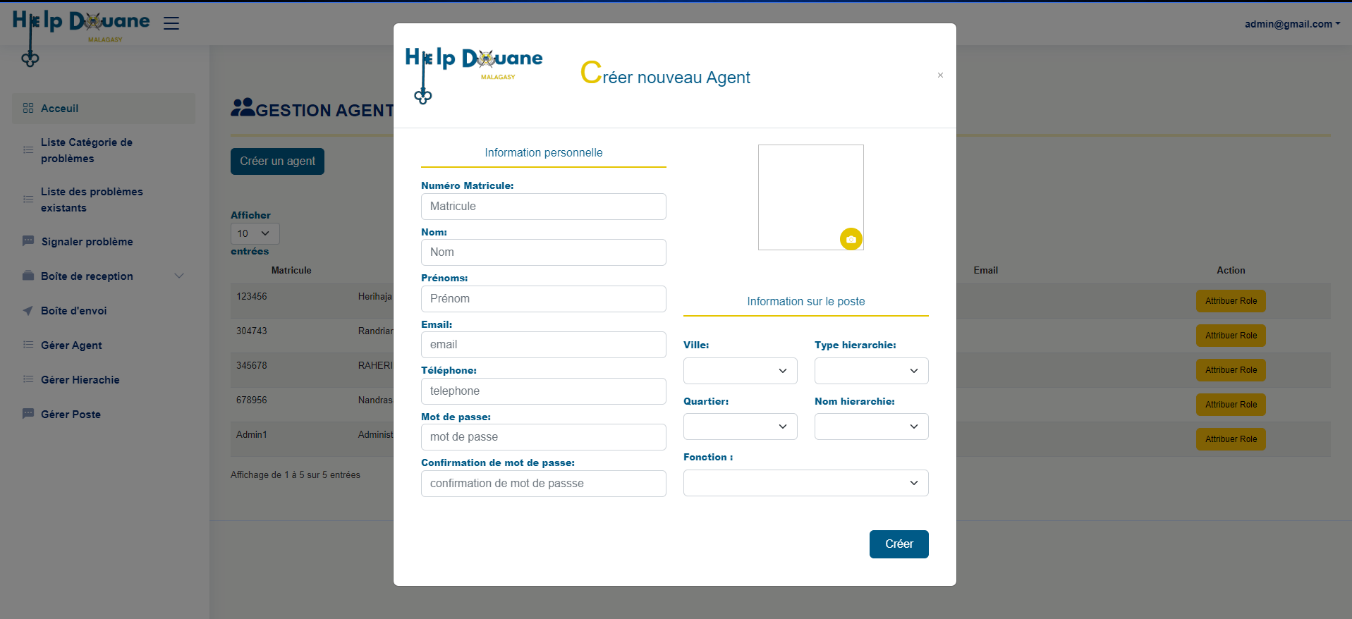


Figure 24 : Page d’auto-enregistrement

8.3.3. Page de listage de demandes

Figure 25 : Page de listage de demandes

Figure : Page Gérer Agent

8.3.4. Page détaillant un emprunteur en demande

#### Page gérer fiche de poste

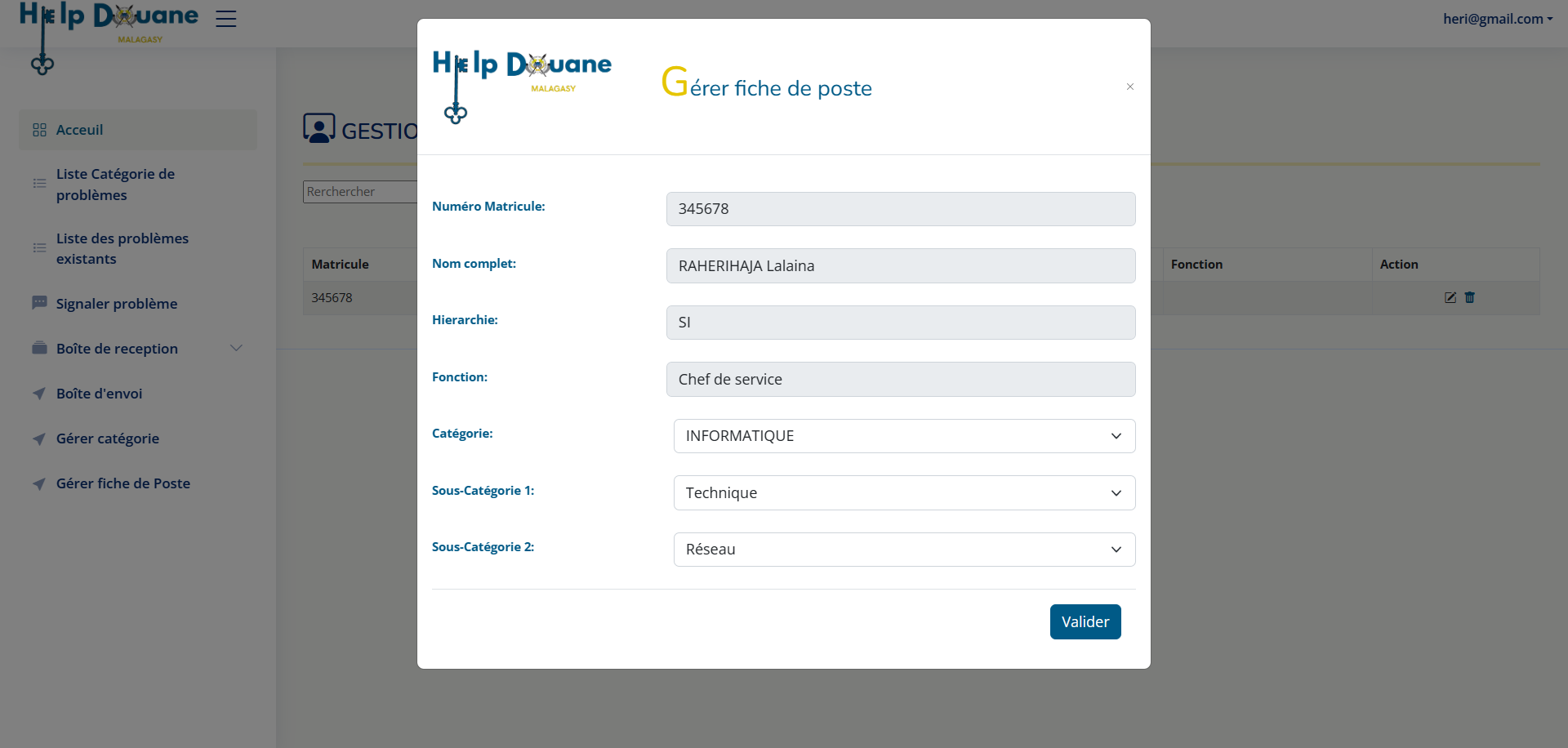


Figure : Page gérer fiche de Poste

#### Page Signaler problème

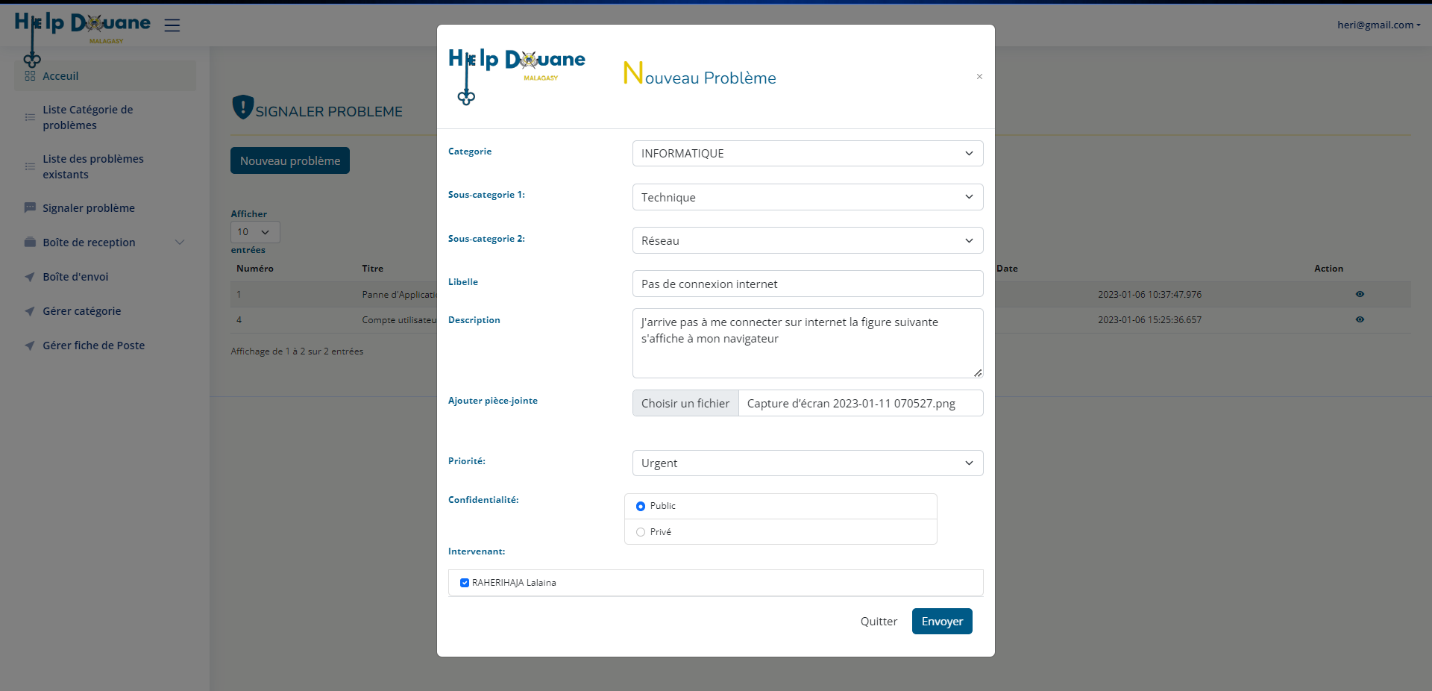


Figure : Page signaler Problème

#### Page de la liste des problèmes existants



Figure : Page consulter Liste des problèmes existants

# CONCLUSION

Pour conclure, ces trois mois de stage étaient une occasion d’enrichir des connaissances professionnelles. De plus l’objectif est de pouvoir entrer dans un monde du métier et d’apprendre à vivre dans une société, communiquer avec plusieurs personnes.

Ce moment occasionnel s’est déroulé au niveau de la douane Malagasy au sein d’une division du service de l’informatique assurant le développement d’autres applications utiles pour les activités de la douane. Le principal but était de mettre en place un genre de système de ticketing à ce niveau, c’est-à-dire dire de concevoir et de réaliser une application capable de gérer les interventions aux problèmes liés à chaque agent.

Pour la conception du projet, la méthode de conception 2TUP avec le langage de modélisation UML est utilisée, l’outil de conception utilisé est le Visual Paradigm, Sur le côté réalisation du projet, le SGBD ORACLE est utilisé avec le langage de programmation java en profitant de son framework Spring Boot, et comme outil de développement, le visual Studio Code est servi.

A la fin du projet, nous avons une application utile pour chaque agent de la douane de demander de l’aide ou de venir en aide à des problèmes, de plus de pouvoir tout de suite voir une solution à ceux qui sont déjà résolus. Elle favorise aussi une orientation facile pour chacun face à un blocage durant son travail.

Le projet répond aux besoins de l’utilisateur malgré quelques améliorations à apporter comme l’implémentation des fonctionnalités capable de faire des appels videos en direct, et aussi d’enregistrer des messages vocaux directement.

# BIBLIOGRAPHIE

[1] CHRISTIAN Soutou, 2013. UML 2 pour les bases de données, 316 pages

[2] Sam NEWMAN, 2015. Microservices Patterns, Etats Unis, Manning, 522 pages

[3] Chris RICHARDSON, 2018. Building Microservices, Etats Unis, O’REILlY, 473 pages

[4] Nate MURRAY, Felipe COURY, Ari LERNER et Carlos TABORDA, 2017. Ng-Book the complete Book on Angular 4, California, Fullstack.io, 681 pages.

# WEBOGRAPHIE

[5] www.openclassroom.com, Tutoriel Java, consulté en Octobre 2022

[6] www.blog.axopen.com, Java ou PHP pour la création d’une application web, consulté en Octobre 2022

[7] https://laurent-audibert.developpez.com, Cours-UML, consulté en Octobre 2022

[8] http://blog.paumard.org, Cours JPA, consulté en Novembre 2022

[9] RAZAFIMAHATRATRA Hajarisena, 2TUP, 2p.

# GLOSSAIRE

**Agent :** une personne qui travaille dans une entreprise.

**Catégorie :** permet de regrouper des problèmes par son genre, définit un classement par nature de problème.

**Dispatcheur :** celui qui réparti toutes les tâches de chaque personne afin de bien distribuer leur responsabilité envers un problème.

**Fiche de poste :** fiche permettant d’identifier les attributions appropriées à une personne de façon plus détaillée et plus complet.

**Hiérarchie :** c’est la structure au niveau de l’organisation d’une entreprise et montre l’ordre de subordination présent dans l’organigramme

**Intervenant :** une personne qui intervient et aide quelqu’un à résoudre un problème.

**Logique métier :** La logique métier fait référence aux processus sous-jacents d'un programme qui effectuent les opérations entre les serveurs d'une entreprise et l'interface utilisateur avec laquelle les clients de cette entreprise interagissent.

**Poste :** représentation de l’occupation d’une personne au niveau d’une entreprise décrivant son rôle et attribution.

**Problème :** une difficulté ou obstacle qui bloque quelqu’un a ne pas accomplir ses tâches ou ses objectifs et qu’il faut résoudre.

**Système de Ticketing :** est un logiciel de gestion, de suivi et de résolution de problèmes rencontrés par la clientèle. Il représente un problème, bug, panne comme un ticket et sert à traiter la priorité et les plus importants à prendre en main.

**Type Hiérarchie :** définit les niveau d’hiérarchies dans une entre souvent le plus haut niveau est la direction.

Table des matières

[CURRICULUM VITAE I](file:///C:\Users\rakot\Downloads\UNIVERSITE%20DE%20FIANARANTSO1.docx#_Toc125097469)

[SOMMAIRE GENERAL II](#_Toc125097470)

[REMERCIEMENTS III](#_Toc125097471)

[LISTE DES FIGURES IV](#_Toc125097472)

[LISTE DES TABLEAUX VI](#_Toc125097473)

[LISTE DES ABREVIATIONS VII](#_Toc125097474)

[INTRODUCTION GENERALE 1](#_Toc125097475)

[PARTIE I : PRESENTATION 2](#_Toc125097476)

[Chapitre 1 : Présentation de l’Ecole Nationale d’Informatique 2](#_Toc125097477)

[1.1. Information d’ordre général 2](#_Toc125097478)

[1.2. Missions et historique 2](#_Toc125097479)

[1.3. Organigramme institutionnel de l’ENI 4](#_Toc125097480)

[1.4. Domaine de spécialisation 6](#_Toc125097481)

[1.5. Architecture des formations pédagogiques 6](#_Toc125097482)

[1.6. Relations de l’ENI avec les entreprises et les organismes 9](#_Toc125097483)

[1.7. Partenariat au niveau international 10](#_Toc125097484)

[1.8. Débouchés professionnels avec des diplômés 12](#_Toc125097485)

[1.9. Ressources humaines 14](#_Toc125097486)

[Chapitre2 : Présentation de la société ou de l’établissement d’accueil 15](#_Toc125097487)

[2.1. Information d’ordre générale 15](#_Toc125097488)

[2.2. Historique 15](#_Toc125097489)

[2.3. Mission 20](#_Toc125097490)

[2.4. Organigramme 20](#_Toc125097491)

[Chapitre 3 : Description du projet 21](#_Toc125097492)

[3.1. Formulation 21](#_Toc125097496)

[3.2. Objectif et besoins de l’utilisateur 21](#_Toc125097497)

[3.2.1. Objectif 21](#_Toc125097498)

[3.2.2. Besoins de l’utilisateur 21](#_Toc125097499)

[3.3. Moyens nécessaires à la réalisation du projet 22](#_Toc125097505)

[3.3.1. Moyens humains 22](#_Toc125097507)

[3.3.2. Moyens matériels 22](#_Toc125097508)

[3.3.3. Moyens logiciels 22](#_Toc125097509)

[3.4. Résultats attendus 23](#_Toc125097510)

[3.5. Chronogramme des activités 23](#_Toc125097518)

[PARTIE II : ANALYSE ET CONCEPTION 2](#_Toc125097519)

[Chapitre 4 : Analyse préalable 24](#_Toc125097520)

[4.1. Analyse de l’existant 24](#_Toc125097525)

[4.1.1. Organisation actuelle 24](#_Toc125097529)

[4.1.2. Inventaire des moyens matériels et logiciels 25](#_Toc125097530)

[4.2. Critique de l’existant 25](#_Toc125097531)

[4.3. Conception avant-projet 26](#_Toc125097532)

[4.3.1. Proposition des solutions 26](#_Toc125097535)

[4.3.2. Choix d’outils pour la conception et la réalisation 27](#_Toc125097536)

[Chapitre 5 : Analyse et Conception 36](#_Toc125097537)

[5.1. Dictionnaire des données 36](#_Toc125097543)

[5.2. Règles de gestion 39](#_Toc125097544)

[5.3. Représentation et spécification des besoins 42](#_Toc125097552)

[5.3.1. Diagramme de cas d’utilisation 42](#_Toc125097553)

[5.3.2. Priorisation des cas d’utilisation 49](#_Toc125097554)

[5.3.3. Spécification des besoins techniques 50](#_Toc125097555)

[5.4. Modélisation du domaine 51](#_Toc125097556)

[5.4.1. Diagramme de séquence système pour chaque cas d’utilisation 51](#_Toc125097557)

[5.4.2. Modèle du domaine 56](#_Toc125097568)

[Chapitre 6 : Conception détaillée 58](#_Toc125097569)

[6.1. Architecture du système 58](#_Toc125097576)

[6.2. Diagramme de séquence de conception pour chaque cas d’utilisation 59](#_Toc125097577)

[6.3. Diagramme de classe de conception pour chaque cas d’utilisation 71](#_Toc125097578)

[6.4. Diagramme de classe de conception global 76](#_Toc125097579)

[6.5. Diagramme de paquetages 78](#_Toc125097580)

[6.6. Diagramme de déploiement 78](#_Toc125097581)

[PARTIE III : REALISATION 79](#_Toc125097582)

[Chapitre 7 : Mise en place de l’environnement de développement 83](#_Toc125097583)

[7.1. Installation et configuration des outils 83](#_Toc125097586)

[7.1.1. Oracle 83](#_Toc125097595)

[7.1.2. Visual Paradigm 84](#_Toc125097596)

[7.1.3. Visual Studio Code 85](#_Toc125097597)

[7.2. Architecture de l’application 86](#_Toc125097598)

[Chapitre 8 : Développement de l’application 87](#_Toc125097599)

[8.1. Création de la base de données 87](#_Toc125097608)

[8.2. Codage de l’application 89](#_Toc125097618)

[8.2.1. Présentation d’un service 89](#_Toc125097619)

[8.2.2. Exemple de Controller Agent 90](#_Toc125097620)

[8.3. Présentation de l’application 92](#_Toc125097621)

[8.3.1. Page d’authentification 92](#_Toc125097622)

[8.3.2. Page gérer Agent 93](#_Toc125097623)

[8.3.3. Page gérer fiche de poste 94](#_Toc125097624)

[8.3.4. Page Signaler problème 94](#_Toc125097625)

[8.3.5. Page de la liste des problèmes existants 95](#_Toc125097626)

[CONCLUSION 96](#_Toc125097627)

[BIBLIOGRAPHIE VIII](#_Toc125097628)

[WEBOGRAPHIE IX](#_Toc125097629)

[GLOSSAIRE X](#_Toc125097630)

[RESUME XV](#_Toc125097631)

[ABSTRACT XV](#_Toc125097632)

# RESUME

En résumé, dans ce présent projet, une application pour gérer les interventions aux problèmes liés à chaque agent de la douane MALAGASY est conçu et réalisé. L’intérêt de ce projet est d’apporter plus d’amélioration au niveau d’assistance d’aide pour chaque agent afin qu’il puisse bien accomplir ses tâches et diminuer la perte de temps à la résolution d’un problème.

La méthode 2TUP avec le langage de modélisation UML a été choisi comme méthode de conception, le langage de programmation Java au profit de son framework Spring Boot comme langage de programmation et à la fois comme environnement de développement le VS code, et ORACLE comme SGBD

Mots clé : 2TUP, Framework, Intervention, problème, SGBD, UML

# ABSTRACT

In summary, in this present project, an application to manage interventions to problems related to each MALAGASY customs agent is designed and produced. The interest of this project is to bring more improvement to the level of assistance for each agent so that he can perform his tasks well and reduce the loss of time when solving a problem.

The 2TUP method with the UML modeling language was chosen as the design method, the Java programming language in favor of its Spring Boot framework as the programming language and both as the VS code development environment, and ORACLE as the DBMS

Keywords: 2TUP, Framework, Intervention, problem, DBMS, UML