UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

MISE EN PLACE D'UNE APPLICATION GESTION DES FORMULAIRES DEMANDE DE VALIDATION DÉROGATION ET DEMANDE ACTION APRÈS CONTRÔLE

LIEU DE STAGE : ATOS

RÉALISÉ PAR : HARTI NACER

SOUTENU LE : 24/06/2015

ENCADRÉ PAR :

MR : Abounaima Med Chaouki

DEVANT LE JURY COMPOSÉ DE :

MR : Abounaima Med Chaouki
MR : Ouzarf Mohamed
MR : Kharroubi Jamal
Mme : Chaker Ilham

ANNÉE UNIVERSITAIRE 2014-2015

Dédicace

Au bon Dieu
Tout Puisant
Source de toute connaissance
Qui a été toujours à côté de moi
Qui ne m'a jamais laissé tomber.

A mes chers parents

Aucun mot, aucune dédicace ne saurait exprimer mon respect, ma considération et l'amour éternel pour les sacrifices que vous avez consentis pour mon instruction et mon bien être.

Votre générosité et votre bonté ont toujours été un exemple pour moi tous.

Trouvez en ce travail le fruit de votre dévouement et l'expression de ma gratitude et mon profond amour.

A mes sœurs et mes proches.

Vous m'avez toujours soutenue durant toutes mes études, je vous souhaite beaucoup de bonheur et de réussite.

A tous mes amis

Je vous aime tous …

Remerciement

Je tiens à remercier toute personne qui m'a aidée à acheminer à bon port le présent Projet de Fin d'Études.

Et bien que ça ne soit l'évidence qui le dicte, je tiens à rendre grâce à mes chers parents, mes sœurs, à toute ma famille qui n'a ménagé aucun effort pour m'épauler, me soutenir, m'encourager et m'aider à venir à terme de cet humble travail.

Qu'il me soit permis d'exprimer mes sincères remerciements à Mr. Mohamed Chaouki Abounaima mon professeur qui m'a orientée tout au long de ce projet.

J'adresse avec tout le respect et l'estime que cela se doit de requérir, mes remerciements à mes enseignants qui ont assuré ma formation tout au long de ma période d'études et Mr. CHLYAH, Jawad le directeur du projet Allianz à Atos qui m'a accueillie au sein du projet.

Résumé

A ce jour c'est plus de 25000 demandes de révision d'assurance qui ont été créées par les équipes de souscripteurs internes d'Allianz, la première demande datant du 15 avril 2009.

Ce volume a induit une demande d'évolutions du système existant pour une meilleure qualité du reporting (qualitatif et quantitatif) et également pour permettre une meilleure maîtrise des risques souscrits.

C'est dans cette perspective que le client Allianz s'oriente vers l'intégration et l'évolution de nouvelles fonctionnalités dans son mode de gestion en une refonte du système existant.

La société se chargeant de la mise en œuvre du processus de gestion est Atos l'une des principales SSII (Société de Services en Ingénierie Informatique).

Pour la réalisation de ce projet j'ai adopté le processus de développement en Y (2TUP) qui permet une meilleure gestion des risques.

Pour la modélisation je me suis basé sur le langage UML.

En ce qui concerne le développement, on a utilisé l'architecture J2EE avec les Framework /Hibernate pour le mapping objet relationnel et JSF pour la couche web.

Abstract

Nowadays, there are more than 25000 requests for insurance review that were created by internal subscriber teams of Allianz, the first request was in 15 April 2009.

This volume induced the evolution request of the existing system for a good quality of reporting (qualitative and quantitative) and to allow a good practice of the subscribed risks.

It is in this perspective that the customer ALLIANZ is moving towards integration and the evolution of new features in its management in an overhaul of the existing system.

The company being responsible for the implementation of the management process is Atos, one of the most important Computer Engineering Services Company.

To accomplish this project I followed the process of development in Y (2tup) which allows a good management of risks.

The UML (Unified Modeling Language) is the technology I used for modeling

As regards the development, I used the J2EE architecture with the framework /Hibernate for the object relational mapping and JSF for the web layer.

Table des matières

Chapitre 1 : Contexte général du projet 10

1. Présentation de l'organisme d'accueil 10
   1.1 Secteurs d'activités 11
   1.2. Organigramme 13
   1.2.1 Atos Casablanca 14
   1.2.2 L'organisme du client 14
2. Contexte du projet 16
3. Objectifs du projet 16
4. Déroulement du stage 16
5. Conclusion 18

Chapitre 2 : Analyse fonctionnelle 19

1. Avant-propos 19
2. Processus de développement 19
   2.1. Choix de la méthode « 2TUP » 19
   2.2. Présentation de la méthode « 2TUP » 19
3. Planning du projet 21
4. Diagramme de GANTT 21
5. Description du système existant 22
6. Périmètre du projet 22
7. Capture des besoins Fonctionnels 22
   7.1. L'authentification 23
   7.2. La consultation 23
   7.3. La modification 23
   7.4. La relance 23
   7.5. La customisation 23
8. Diagramme de cas d'utilisation 24
9. Cycle de vie d'une demande 28
10. Conclusion 29

Chapitre 3 : Analyse Technique 30

1. Avant-propos 30
2. Les différents états d'une Demande : 30
   2.1. Demande non validée/ en modification 30
   2.2. Demande validée 30
   2.3. Demande refusée 30
   2.4. Demande Sans suite : 30
   2.5. Demande en transfert 30
   2.6. Demande en relance 31

3. Diagramme d'état 31
4. Diagrammes de séquences 32
5. Architecture physique 34
6. Architecture logique 35
7. Conclusion : 36
Chapitre 4 : CONCEPTION 37
1. Avant-propos 37
2. Identification des classes candidates 37
2.1. Diagramme de classe 38
3. Description des packages et structure programme 38
4. Conclusion 40
Chapitre 5 : mise en œuvre 41
1. Avant-propos 41
2. Outils de développement 41
2.1 Environnement de développement Eclipse 41
2.2 Serveur web et d'application : Apache Tomcat 41
2.3 WampServer : 41
2.4 Entreprise Architect : 41
3. Framework de développement : 42
4. Interfaces Graphiques 42
4.1. Interface d'accueil 42
4.2. Interfaces de gestion des composants dynamique 43
4.3. Interface gestion des Utilisateurs 44
4.4. Interface de gestion des catégories de risques 44
4.5. Interface de gestion des profiles 45
4.6. Interface de gestion des motifs de demande 45
4.7. Interface de gestion des Types affaires 46
4.8. Interface de consultation des demandes 46
4.9. Interface de saisie d'une demande 47
4.10. Interface de redirection de la saisie de demande 48
4.11. Interfaces de consultation demande 48
4.12. Interface du choix de validation demande 49
5. Conclusion 49
Conclusion général 50
Bibliographie 51
1. Les ouvrages 51
2. Les sites-web 51
3. Forums 51

Liste des Figures

Figure 01 Carte des pays de localisation des services au niveau du groupe Atos……..…11
Figure 02 Les activités d'Atos…….. ..12
Figure 03 Organigramme d'Atos…….....13
Figure 04 Organigramme d'Atos Casablanca……..14
Figure 05 Le système d'information soumis à deux natures de contrainte……..15
Figure 06 Le processus de développement en Y…….19
Figure 07 Le Diagramme de GANTT……...…..21
Figure 08 Diagramme de cas d'utilisation du validateur et souscripteur…….…...…25
Figure 09. Diagramme de cas d'utilisation de l'Administrateur……..27
Figure 10. Diagramme d'état de transition d'une demande……..31
Figure 11 Diagramme de séquence du chargement page accueil……...…32
Figure 12. Diagramme de séquence du traitement d'une demande……..33
Figure 13. Diagramme de séquence de saisie demande……....34
Figure 14 Architecture physique du projet……..35
Figure 15 L'architecture en couche de notre application……...…….36
Figure 16 Diagramme de classe du Système……..….38
Figure 17 structure en package du système…….....39
Figure 18 Interface d'accueil
Figure 19 Interfaces de gestion des composants dynamique……...**Erreur ! Signet non défini.**
Figure 20 Interface gestion des Utilisateurs .....
Figure 21 Interface de gestion des catégories de risques .....
Figure 22 Interface de gestion des profiles .....**Erreur ! Signet non défini.**
Figure 23 Interface de gestion des motifs de demande .....
Figure 24 Interface de gestion des Types affaires.....**Erreur ! Signet non défini.**
Figure 25 Interface de consultation des demandes.....
Figure 26 Interface de saisie d'une demande .....**Erreur ! Signet non défini.**
Figure 27 Interface de redirection de la saisie de demande .....
Figure 28 Interfaces de reprise demande pour validation .....
Figure 29 Interface du choix de validation demande……...

Liste des Abréviations

Abréviation	Désignations
J2EE	Java Enterprise Edition
COBOL	Common Business Oriented Language
TMA	Tiers Maintenance Applicative
2TUP	2 Track Unified process
SSII	Société de Services en Ingénierie Informatique
ASP	Active Server Pages
HQL	Hibernate Query Language
DVD	Demande de validation dérogation
DAC	demande action après Contrôle
MVC	Modèle-Vue-Contrôleur
MVS	Multiple Virtual Storage
UML	Unified Modeling Language
JSF	Java Server Faces
JSP	Java Server Pages
IHM	Interface Homme-Machine
DAO	Data Access Object
CFD	Conception Fonctionnelle Détaillée
CTD	Conception Technique Détaillée
SSL	Secure Sockets Layer

Introduction générale:

Suite aux demandes issues des retours utilisateurs et des équipes de contrôles internes d'Allianz (le client) une refonte du programme existe pour la gestion des demandes de dérogation a été sollicitée pour un meilleur traitement qualitatif et quantitatif.

Mon travail consiste en la création et la mise en place d'une application web permettant la gestion des formulaires DVD (demande de validation de dérogation) et DAC (demande action après Contrôle).

Mettre en forme puis de faire transiter des informations liées à une souscription vers le validateur (la personne qui va donner suite à la demande) idoine et validateur en copie en leur adressant les demandes par courrier électronique et avec un Système de relance ainsi qu'une gestion dynamique et personnalisable des demandes.

Ce travail est divisé sur 5 chapitres :

- Le 1èr chapitre décrira le contexte général du projet qui comporte une brève présentation de la société, la problématique et la solution proposée, ainsi que les différentes tâches réalisées au cours de mon stage.
- Le 2ème chapitre définira le processus de développement adopté et décrira l'analyse fonctionnelle qui définira les besoins et détaillera les aspects fonctionnels du système.
- Le 3ème chapitre décrira l'analyse technique qui définira les besoins et les aspects techniques et la mise en place de l'architecture de l'application.
- Le 4éme chapitre comportera la partie conception et réalisation qui définira le modèle de conception du système.
- Le 5éme chapitre présentera les outils de développement et les technologies choisies et aussi un aperçu de l'application.

Chapitre 1 : Contexte général du projet

1. Présentation de l'organisme d'accueil

Ce projet de fin d'étude a comme espace de travail le centre de services d'Atos à Casablanca, l'une des principales SSII françaises leader de services numériques avec un chiffre d'affaires annuel de 10 milliards d'euros et 86 000 collaborateurs dans 66 pays.

Atos se décrivant comme l'un des principaux acteurs internationaux dans les services informatiques en faisant partie du Top 10 mondial et en se classant no 5 mondial en Infogérance. Elle est leader en Europe parmi les acteurs européens et leader du paiement sécurisé en ligne pour les entreprises en France. Grâce à cette opération, Atos renforce son engagement en matière d'innovation et accélère sa stratégie de croissance dans le domaine des services transactionnels.

Atos délivre les technologies qui accélèrent le développement de ses clients et les aident à réaliser leur vision de l'entreprise du futur. En tant que société responsable et intégrée, Atos s'est engagé à décliner les meilleures pratiques dans les domaines environnementaux, sociaux et éthiques au sein de toute son organisation. Le Groupe contribue ainsi à promouvoir et à développer un comportement responsable en influençant positivement ses parties prenantes pour qu'elles prennent en compte le développement durable dans leurs processus de décision. Atos aide ses clients à avancer dans ces domaines, à réduire leur empreinte environnementale et à assurer leur viabilité future en leur fournissant des solutions innovantes et durables.

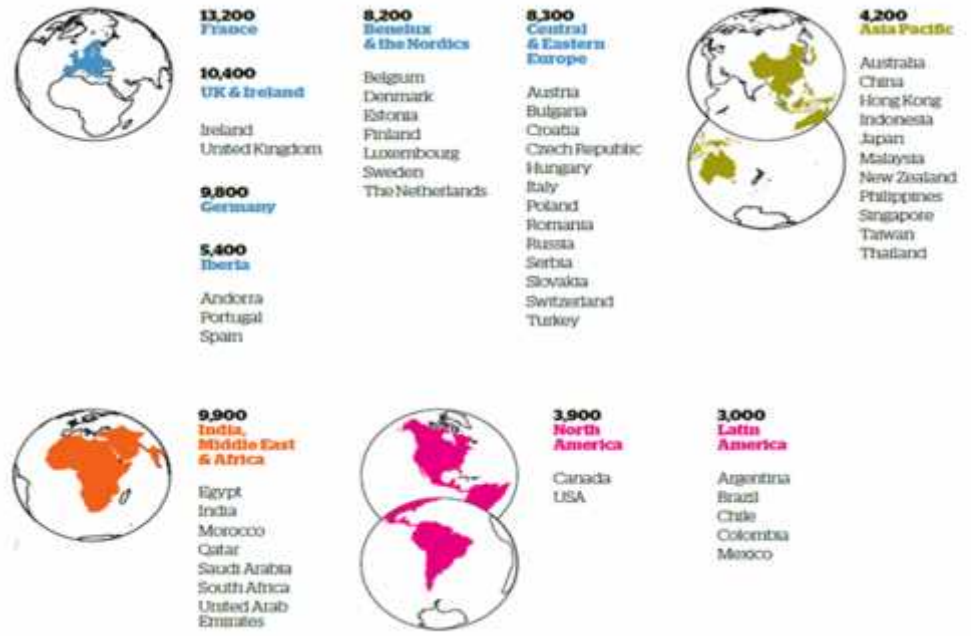


Figure 01. Carte des pays de localisation des services au niveau du groupe Atos

1.1 Secteurs d'activités

La concentration d'Atos sur ses clients, le partage de fortes valeurs et l'implication de ses collaborateurs, sont à la base de son succès. Le groupe ne cesse de penser aux besoins et demandes de ses clients, ainsi on arrive à trouver une grande diversité dans ses activités qui peuvent se résumer en 4 métiers principaux :

- Le conseil et les services technologiques
- L'intégration de systèmes
- L'infogérance
- Les services transactionnels de haute technologie (HTTS)

Figure 02. Les activités d'Atos.

1.1.1. Atos Consulting

Atos Consulting s'assure que tous les composants d'une organisation cliente : les ressources humaines, les nouvelles technologies et les flux d'investissements suivent la stratégie de l'entreprise. Le Groupe accompagne ses clients à chaque étape du processus de changement depuis sa planification stratégique jusqu'à sa mise en œuvre, son exploitation et le management de ses ressources humaines.

1.1.2. Atos Intégration

Elle ne consiste pas uniquement à faire une intégration de nouvelles solutions, mais également d'en tirer le meilleur, afin de rentabiliser l'investissement. En assurant une association maîtrisée de nouvelles solutions aux processus existants, elle aide à transformer l'architecture complète de l'entreprise en un système opérationnel et uniforme. Leurs spécialistes travaillent avec ses clients au développement, à la mise en œuvre et à la maintenance de systèmes conçus pour prendre en charge leur stratégie d'entreprise dans son ensemble.

Elle collabore avec des partenaires stratégiques soigneusement sélectionnés en vue de développer et mettre en œuvre des offres complètes et des solutions logicielles dans des environnements complexes utilisant les technologies les plus avancées du marché. Elle réalise

également des projets Open Source ainsi que des logiciels spécifiques reposant notamment sur des langages et des méthodes de conception différents.

1.1.3 Atos Infogérance

La ligne de service Infogérance assure la gestion totale des infrastructures informatiques de ses clients, incluant les centres, la gestion de parcs informatiques, les serveurs et les systèmes de communication réseau. Le groupe fournit une assistance 7j/7 et 24h/24 à travers leur réseau international. Atos a su profiter de la croissance de l'infogérance en Europe ces dernières années à travers :

- Des extensions des contrats sur les clients existants.
- Favorisation de la demande pour le service Cloud.
- Une croissance significative de la migration d'environnements virtuels comme
- moyen de réduction des couts et de flexibilité.
- Une augmentation de l'externalisation informatique en offshore.
- Insertion parmi les organisations mettant en priorité les exigences environnementales.

1.1.4 AtosWorldline

Atos Worldline est le centre chargé du traitement des services transactionnels de haute technologie. Il est spécialisé dans les paiements électroniques (émission, acquisition, terminaux, solutions de paiement et traitement de cartes), les ECS (services en ligne pour les clients, les citoyens et les communautés), ainsi que les services pour les marchés financiers, il permet à ses clients de bénéficier de solutions innovantes récompensées par le marché dans des domaines tels que le paiement sur mobile, l'IPTV, le CRM en ligne, ou encore les solutions de dématérialisation.

De plus, l'entreprise possède un partenariat exclusif avec le Comité International Olympique (CIO). Ce contrat constitue le plus important contrat informatique lié au monde du sport jamais signé, couvrant six Olympiades : Salt Lake City 2002, Athènes, Turin 2006, Pékin 2008, Vancouver 2010 et Londres 2012.

Atos a la responsabilité principale de l'informatique, en matière de conseil, d'intégration des systèmes, de gestion des opérations, de sécurité de l'information et de développement des logiciels applicatifs pour les Jeux olympiques.

1.2. Organigramme

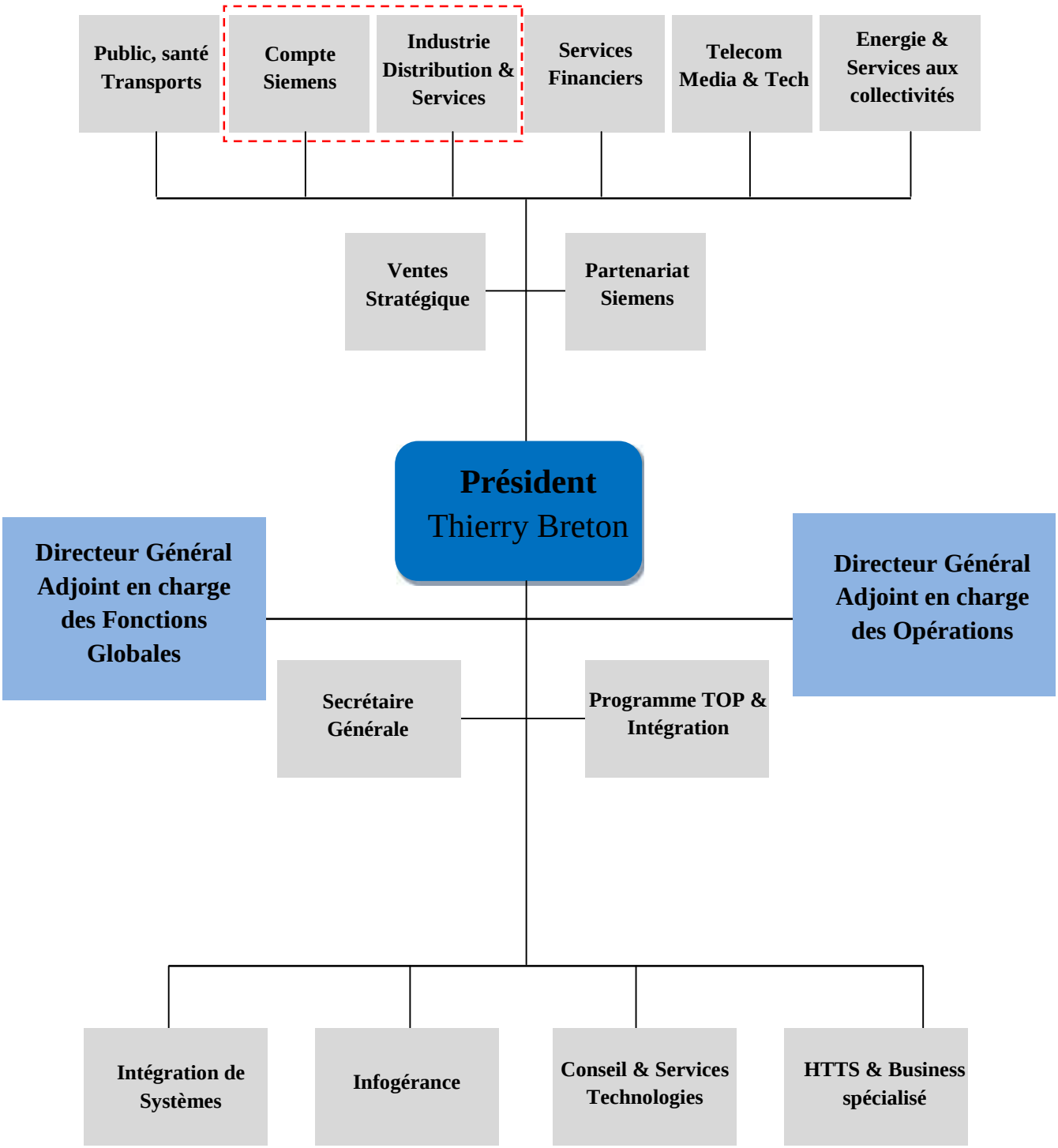


Figure 3. Organigramme d'Atos

1.2.1 Atos Casablanca

Présente depuis 2003 au Maroc, Atos a su établir des relations de partenariat durable avec les acteurs locaux. Depuis 2007 les centres de services de Casablanca, orientés vers l'international et en pleine expansion, fournissent des prestations en intégration de systèmes et infogérance en s'adressant à une clientèle francophones (France, Belgique, Suisse), anglophone ou hispanophone (Espagne) dans les secteurs financiers, les télécommunications, le secteur publique, l'industrie, l'automobile et la distribution.

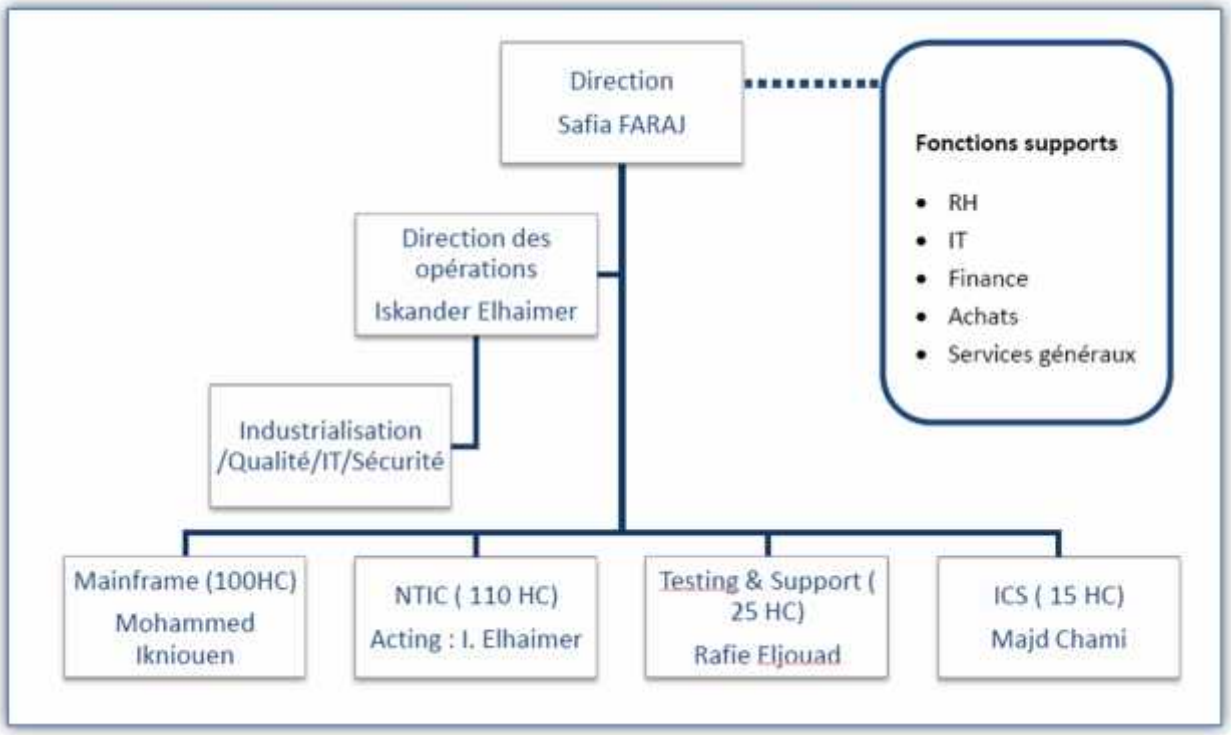


Figure 4. Organigramme d'Atos Casablanca

1.2.2 L'organisme du client

1.2.2.1 ALLIANZ

Allianz SE, créé en 1890 à Berlin, est un groupe d'assurances, le premier assureur européen, et le troisième gestionnaire d'actifs au monde.

Le groupe est diversifié entre l'assurance-dommage et l'assurance-vie, et est actif en Allemagne, en France et en Italie. Le groupe utilise également son réseau d'agents d'assurance pour vendre des produits bancaires.

En février 2007, Allianz acquiert 100 % du capital d'AGF. Le groupe est l'ancien propriétaire de la Dresdner Bank en Allemagne, cédée à Commerzbank en septembre 2008 pour 9.8 MM €.

1.2.2.2 L’assurance

Une **assurance** est un service qui fournit une prestation lors de la survenance d'un événement incertain et aléatoire souvent appelé "risque". La prestation, généralement financière, peut être destinée à un individu, une association ou une entreprise, en échange de la perception d'une cotisation ou prime.

Par extension, l'assurance est le secteur économique qui regroupe les activités de conception, de production et de commercialisation de ce type de service.

L'**assurance** est une opération par laquelle une personne (l'assureur) s'engage à réaliser une prestation, dans le cadre d'un contrat d'assurance, au profit d'un autre individu (l'assuré) lors de la survenance d'un risque et moyennant le paiement d'une cotisation ou d'une prime. Le risque constitue l'objet de l'assurance. Un assuré contracte une assurance pour se prémunir contre des événements dommageables (maladie, incendie, vol, décès...) ou contre les risques liés à certains objets qu'il possède (automobile, habitation...).

En cas de survenance d'un risque, l'assureur s'oblige à verser une prestation sous forme d'argent soit à l'assuré, soit à un tiers, soit au bénéficiaire dans le cadre d'une assurance-vie. En contrepartie de cette prestation, l'assuré verse une prime ou une cotisation à l'assureur. Cette somme correspond au coût du risque et aux frais de fonctionnement de l'assureur. Une assurance doit être souscrite auprès d'une société qui est habilitée à exercer cette activité. Il peut s'agir d'une compagnie d'assurances ou encore d'une banque. La souscription s'effectue par la signature d'un contrat d'assurance.

1.2.2.3 Réassurance

L’assuré souscrit une police auprès d’un assureur direct qui se réassure en cédant une partie des risques à un ou plusieurs **réassureurs**.

L’assureur direct est alors appelé **cédant** et le réassureur **cessionnaire**

Contrat de réassurance passé entre deux assureurs dont l’un accepte une part des risques assurés par l’autre en échange d’une part de la prime.

La réassurance peut se faire selon plusieurs modes d’engagement :

La réassurance est dite **facultative** si elle se négocie affaire par affaire à la convenance des deux parties

La réassurance est dite **obligatoire** quand les deux parties signent un **traité** qui les engage pour une certaine catégorie de risque à venir, l’assureur direct étant tenu d’en céder une partie au réassureur qui est tenu de l’accepter.

La réassurance est dite **facultative-obligatoire** (ou open-cover) si elle est facultative pour l’assureur direct mais obligatoire pour le réassureur.

2. Contexte du projet

Mon stage s'est déroulé au sein de la branche « Système Intégration », celle qui compte le plus de collaborateurs en France. Cette branche a pour vocation d'imaginer, de développer, de maintenir les systèmes d'informations des entreprises et plus précisément pour le client Allianz.

Ce Projet de Fin d'Etudes s'inscrit dans le cadre d'un Système d'information pour la gestion des demandes d'assurance permettant la saisie, le traitement dynamique des demandes en assurant un partage , une alerte par email , un système de relance et la possibilité d'effectuer des extractions des bases de donnée pour une partie exploitation statistique en J2EE.

3. Objectifs du projet

Les principaux objectifs à atteindre par le présent projet de fin d'études sont :

- Mettre en forme et faire transiter des informations liées à une souscription.
- Permettre aux souscripteurs de donner une information complète sur une affaire avant validation par un validateur.
- Dématérialiser les demandes pour assurer une diffusion immédiate par email.
- Un stockage et une sécurisation tant dans les échanges que dans la pérennité de l'information.
- Une exploitation plus ciblée des données récoltées lors des demandes en conformité.
- Un système de relance pour les demandes qui n'ont pas été traitées à j-2 de la date limite de la réponse souhaitée.
- Un système de relance manuel.

4. Déroulement du stage

Le projet a été réalisé en parallèle avec la réponse aux demandes client.

J'ai pu découvrir au cours de mon stage la démarche de travail d'une TMA en travaillant en équipes sur des applications développées dans la technologie ASP.

Mon rôle au sein de la TMA-ASP a été de participé au suivi du processus des demandes, à savoir corriger des anomalies, réaliser de petites évolutions, changement de données, extraction.

Pour la gestion des demandes, on utilise un logiciel nommé APPLIX qui contient toute les informations d'une demande (la description, le type, niveaux de criticité, l'émetteur, ….).

Le client peut commander une évolution, faire part d'une anomalie qu'il a pu rencontrer ou de demander une modification sur les bases de production.

Dans ces cas, une personne l'équipe de la TMA ASP prend en charge la demande, effectue une analyse des spécifications de la CFD, rédige la CTD avec une description de la solution technique, les fonctionnalités à implémenter puis le développement.

Dans ce cadre, on utilise un outil de développement nommé Microsoft Expression Web.

Avant d'exécuter la phase de qualification, la personne en charge :

- livre en recette les modules modifiés.
- Rédige le bon de livraison avec les pages modifiées (asp), les scripts des requêtes base de données, les batch (au cas d'une modification d'un batch nuit, batch extraction (delphi).
- sollicitation auprès de la DP (direction du pilotage, au cas d'un manque d'habilitation lors d'exécution d'une requête ou copie du contenu d'un répertoire de la production vers la recette…).
- mise en production (après validation),

Au cours de mon stage on a pu traiter les demandes suivantes :

- Problème qui bloque l'utilisateur lors de la saisie des demandes, après analyse, il s'est avéré qu'il s'agit d'un problème de la durée de session, donc on a dû chercher les pages où le timeout de la session est paramétré, quel est le timeout (la durée de vie) par défaut 20min puis une analyse sur les pages à modifier pour que toute la plateforme ai un nouveau timeout plus long.
- Dans le cadre de la migration du système d'exploitation et le changement des versions de Microsoft office, un site 'sésame' que les salariés d'Allianz consulte pour toute information, propose un ensemble de liens qui pointent sur des fichiers office, Excel,...

  L'ouverture de ces derniers utilise une authentification de base, alors qu'OFFICE 2010 exige la mise en place du SSL.

  Dans ce cas, on a proposé un passage à https.
- Le paramétrage des modules du site sésame Admin est totalement dynamique et généré à travers le choix d'un fichier qui contient les modules.

  Ces derniers sont générés avec des critères de tri qui présente une anomalie.

  Après analyse, l'anomalie résulte du format différent des dates saisies dans le
- Problème d'insertion simultanée des enregistrements, avec le même identifiant, un problème difficile à reproduire, toute fois il fallait une solution implémenter les verrous au niveau de la base de données.

- Une analyse pour la migration du navigateur internet explorer 8 au navigateur Internet explorer11.
- tests techniques, au niveau des composants (HTML, CSS, VB script) détecte les pages non affectées en mode natif, ceux qui nécessite le mode compatibilité et ceux qui nécessite une évolution.
- Une formation en COBOL et l'environnement MVS.

5. Conclusion

Dans ce chapitre, on a présenté l'organisme d'accueil Atos et ses domaines d'activités. On a aussi présenté d'une manière générale le contexte, le déroulement de mon stage ainsi que les objectifs à atteindre par ce Projet de Fin d'Etudes.

Chapitre 2 : Analyse fonctionnelle

1. Avant-propos

L'étude de l'existant est une étape clé dans la réalisation de n'importe quel système d'information, quel que soit le domaine concerné. Il s'agit d'une étude permettant de comprendre la problématique du projet au cours de ce chapitre nous allons détailler les besoin fonctionnels du system et le modèle de développement adopter.

2. Processus de développement

2.1. Choix de la méthode « 2TUP »

On a opté pour la méthode 2TUP car elle est plus adaptée à la structure et à l'environnement de mon projet. Elle apporte des solutions aux quatre risques majeurs ci-dessous d'un projet:

- L'inadéquation aux besoins des utilisateurs.
- L'incapacité de l'architecture technique à répondre aux contraintes opérationnelles.
- Le non-respect des délais.
- Le manque de qualité.

2.2. Présentation de la méthode « 2TUP »

2TUP signifie « 2 Track Unified Process ». C'est un processus UP qui répond aux caractéristiques citées ci-dessus. Le processus 2TUP apporte une réponse aux contraintes de changement continuel imposées aux systèmes d'information de l'entreprise. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes des changements imposés au système informatique.

Figure 5: Le système d'information soumis à deux natures de contrainte

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion conduit à l'obtention d'un processus de développement en forme de Y, comme illustré par la figure 6 :

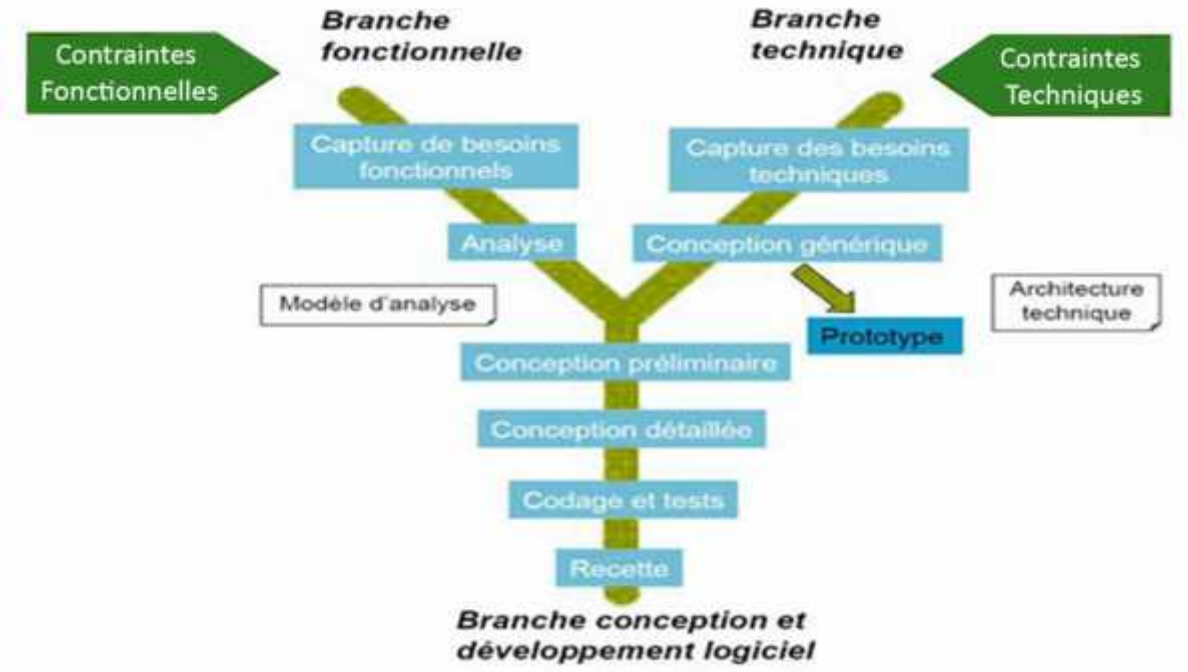


Figure 6 : Le processus de développement en Y

La branche gauche (fonctionnelle) comporte :

- la capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs.

La branche droite (architecture technique) comporte :

- la capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant.
- la conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique.

La branche du milieu comporte :

- la conception préliminaire, qui intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- la conception détaillée, qui étudie ensuite comment réaliser chaque composant.

- l’étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code.
- l’étape de recette, qui consiste enfin à valider les fonctions du système développé.

3. Planning du projet

Une phase de planification s’avère primordiale afin d’assurer la bonne conduite du projet puisqu'elle participe au succès de la démarche.

En effet, la planification et l’ordonnancement sont des phases essentielles du projet. Il s’agit d'identifier le meilleur enchaînement des tâches indispensables à la réalisation du projet. Ainsi la planification est un outil incontournable de la gestion du projet permettant de :

- Définir les travaux à réaliser.
- Fixer les objectifs.
- Coordonner les actions.
- Diminuer les risques (Dépassement des délais).

4. Diagramme de GANTT

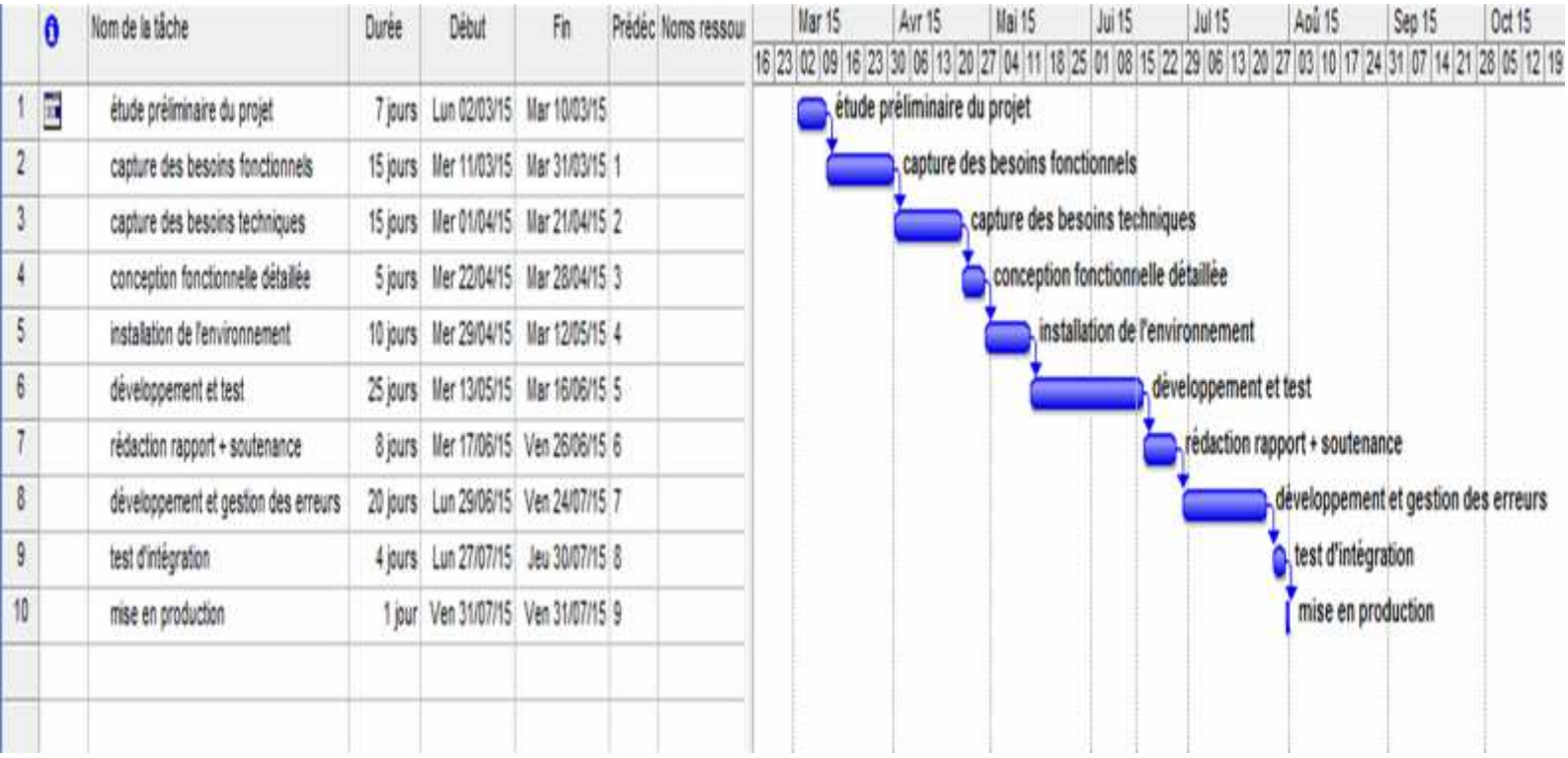


Figure 7 : Le Diagramme de GANTT

5. Description du système existant

Pour gérer les demandes d'assurance le système existant assurait la saisie et le partage des demandes sous une plateforme web programmée en ASP classique.

Critique de l'existant :

Modification lourde à effectuer :

- Les reprises sur les programmes demande une grande analyse vue la nature du langage.
- Demandes non précises :
  Le formulaire manque de détaille pour un traitement optimiser des demandes.
- Difficulté d'avoir l'information avec rapidité.
  Les critères de recherches et d'extractions peu précises.
- Un contrôle faible sur la saisie des demandes.
  Les souscripteurs peuvent saisir des valeurs absentes dans les bases de données.
- Difficulté d'analyse de l'information, temps de traitement de l'information est important.

6. Périmètre du projet

Le projet vient pour répondre au besoin de l'évolution d'un système d'information permettant la gestion des demandes d'assurance avec des formulaires plus ciblés et avec un langage évolué et plus facile à maintenir.

Pour réaliser ce projet, les missions suivantes sont indispensables :

- Elaborer l'étude préalable.
- Elaborer l'étude fonctionnelle.
- Elaborer l'étude technique.
- Analyse et conception.
- Réalisation, intégration et documentation.

7. Capture des besoins Fonctionnels

Dans l'optique de concevoir une application qui soit modulaire, facilement extensible et orientée objet, le formalisme UML s'est imposé comme l'outil le plus approprié pour ce projet. En effet, UML permet de mener la phase d'analyse et de conception tout en bénéficiant de la

puissance et de la simplicité de l'orienté objet UML est également le formalisme adapté au processus de développement choisi :

La capture des besoins fonctionnels a pour objectifs :

- L'identification des cas d'utilisation du système par ses acteurs.
- La description des cas d'utilisation.
- L'identification des classes candidates des modèles d'analyse.
- La description des états de transition d'une demande.

7.1. L'authentification

Une phase d'authentification est nécessaire pour pouvoir accéder aux fonctionnalités de l'application. Une seule authentification (Single Sign On) suffit pour permettre à l'utilisateur d'accéder aux fonctionnalités de l'application requise, cette authentification est effectuée en récupérant les propriétés Systèmes.

Seul l'administrateur aura le droit d'accéder à toutes les fonctionnalités de l'application.

Les utilisateurs hors périmètre seront redirigés vers une page d'erreur.

7.2. La consultation

Les Utilisateurs peuvent accéder à toute demande par le biller de l'historique, la saisie du numéro de la demande, recherche par critères ou à travers un lien direct.

7.3. La modification

Les validateurs peuvent reprendre et modifier le statut de la demande s'il est diffèrent d'un état final (valider ou Sans suite) ou en relance.

Les souscripteurs peuvent reprendre et modifier une demande sans changer son état.

7.4. La relance

Les utilisateurs peuvent effectuer une relance manuelle pour accélérer le traitement d'une demande.

7.5. La customisation

L 'Administrateur peut ajouter, modifier, supprimer les éléments du formulaire et de modifier l'ordre d'affichage et désactiver des champs ou des valeurs au choix.

8. Diagramme de cas d’utilisation

Les diagrammes de cas d'utilisation sont utilisés pour donner une vision globale du comportement fonctionnel d’un système logiciel.

Avant d’entamer l’illustration des diagrammes de cas d’utilisation, on définit les principaux acteurs de notre système :

Souscripteur : le front office prend en charge la récolte des informations d’une demande.

Validateur : le back office prend en charge l’étude et le traitement d’une demande.

Administrateur : Prend en charge la gestion de la plateforme et la création des utilisateurs et l’attribution des profils.

Le diagramme de cas d’utilisation des acteurs (validateur et souscripteur) Figure 8 représente toutes leurs interactions avec le système.

Le souscripteur peut créer, consulter, modifier, compléter des informations sur une demande quel soit DVD ou DAC et d’effectuer une relance du validateur principale.

Le validateur est un souscripteur qui peut modifier l’état de la demande (traiter la demande).

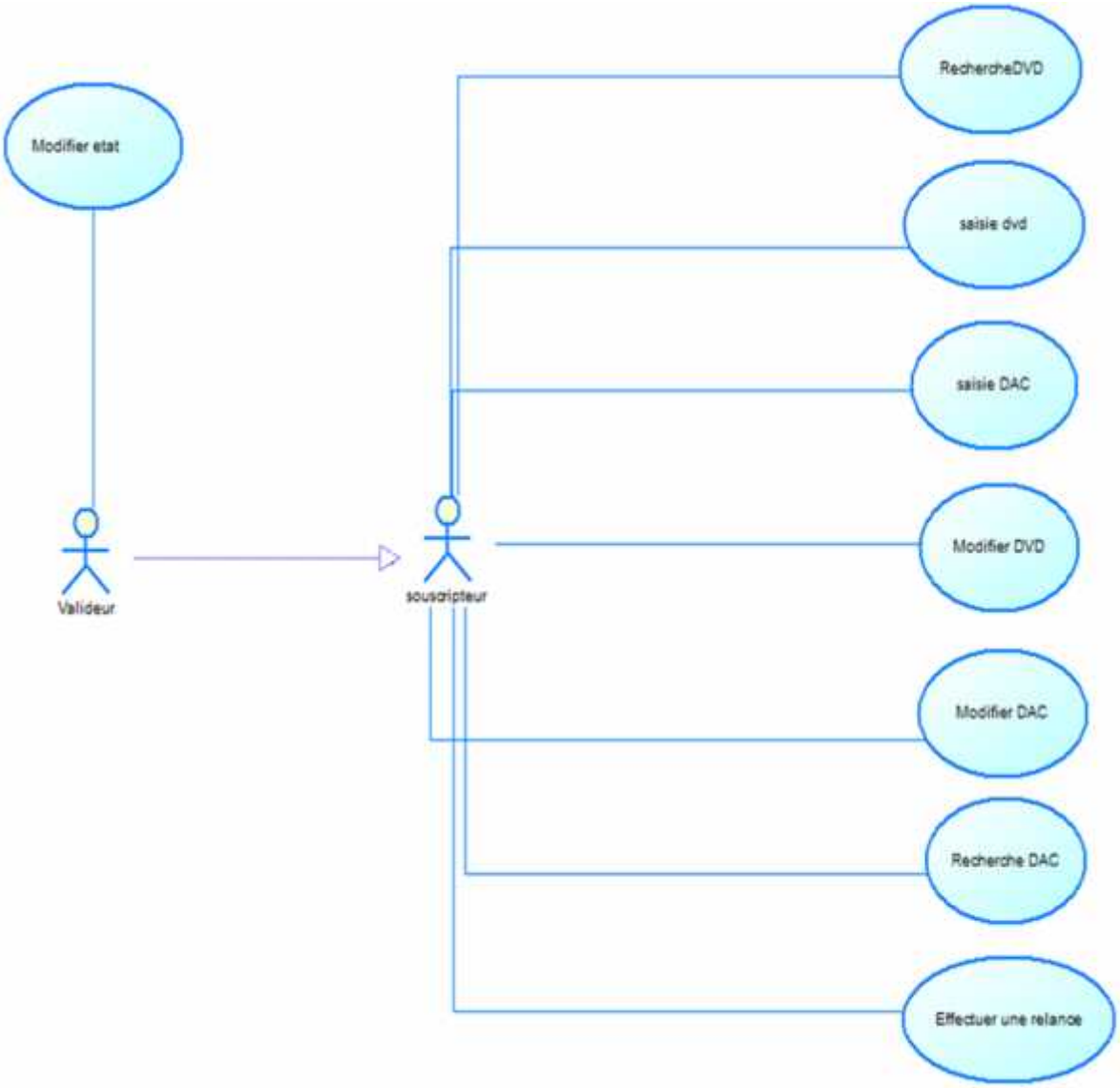


Figure 8. Diagramme de cas d'utilisation du validateur et souscripteur

Le diagramme de cas d'utilisation de l'administrateur (Figure 9) : représente tous ces interactions avec le système.

L'administrateur est un validateur, qui peut effectuer l'ajout, la modification, la suppression des demandes et des objets dynamiques du formulaire et organiser leur ordre d'affichage :

- Fonction Cartouche Émetteur
- Équipe de souscription
- Type d'Affaire
- Enregistrement Informatique
- Marché
- Numéro de Saisine ou d'Étude ou de Contrat
- Canal de distribution
- Code et libellé TAIRE

- Code et libellé NAF V2003
- Code et libellé Construction
- Catégorie du Risque
- Mode de Gestion
- Motif de la demande
- Gestion de la sélection du validateur principal et/ou copies
- Traitement de la demande

L’administrateur doit aussi pouvoir effectuer des extractions de la base de données sous format CSV ou Excel et doit avoir le droit de suppression d’une demande.

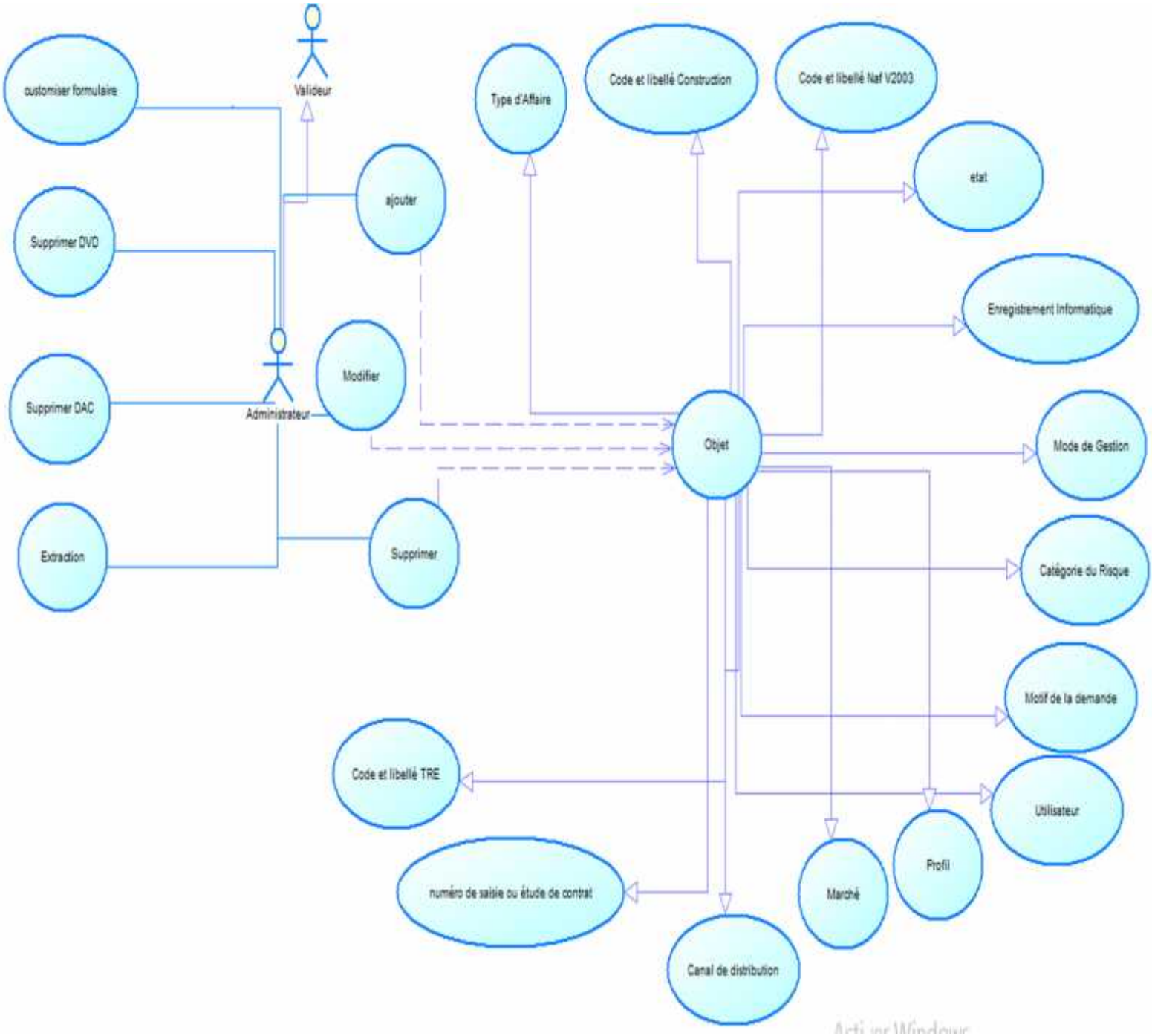


Figure 09. Diagramme de cas d'utilisation de l'Administrateur

9. Cycle de vie d'une demande

Émetteur	Validateur 1	Validateur n+1	Validateur final
Saisie du formulaire			
	Transfert Envoi du courrier A : V1 Cc : Émetteur	Transfert Envoi du courrier A : V.n+1 Cc : Émetteur ; V1	Sans suite ou Refus Envoi du courrier A : V. final Cc : Émetteur ; V1 ; V.n+1
		Sans suite ou Refus Envoi du courriel A : dernier émetteur CC : Émetteur et toutes partie des V.n+1	
	Sans suite ou Refus Envoi du courriel A : dernier émetteur CC : Émetteur et toute partie des V.n+1		
Sans suite ou Refus Envoi du courriel A : Émetteur CC : Valideur 1			Validation Envoi Demande aux destinataires dont émetteur et validateurs
	Validation Envoi Demandeaux destinataires dont émetteur et validateurs	Validation Envoi Demandeaux destinataires dont émetteur et validateurs	

10. Conclusion

La description détaillée du système existant réalisée dans le début de ce chapitre a permis de délimiter les problèmes à résoudre dans le contexte actuel. Quant à la partie « critique de l'existant », elle a permis de dégager et de formaliser les besoins des futurs utilisateurs du système. Après avoir déterminé le contexte, les objectifs, la méthodologie du travail, le planning, les tâches du projet, le chapitre suivant concerne l'étude préalable du projet.

Chapitre 3 : Analyse Technique

1. Avant-propos

Conformément à la démarche choisie, cette partie sera consacrée à la branche technique du projet. En fait, le choix de l'architecture technique s'appuie sur les objectifs liés aux besoins techniques de l'application.

2. Les différents états d'une Demande :

Le formulaire a toujours un état et un seul, quand la Demande est en état « refusée » mais reprise pour une nouvelle validation il faut que l'état « refusée » reste présent dans le formulaire tant qu'un nouvel état n'est pas initialisé par un validateur final.

2.1. Demande non validée/ en modification

Une demande est en état non validé quand elle est en cours de constitution ou en attente de validation.

2.2. Demande validée

Une DVD est en état validé quand le validateur principal ou secondaire fait ce choix et le valide par le bouton.

2.3. Demande refusée

Une demande refusée peut-être reprise par son émetteur. Elle perd donc son statut refusé quand elle est renvoyée à un validateur (mais l'information doit rester dans le corps du formulaire, tant que son nouvel état n'est pas précisé).

2.4. Demande Sans suite :

Une Demande en état « Sans Suite », est une demande qui n'a pas forcement suivi le cycle de validation, à savoir que l'émetteur peut lui-même passer sa Demande en sans suite.

Une demande « Sans suite »ne peut plus être reprise (modifiée).

2.5. Demande en transfert

Il s'agit d'une fonctionnalité qui permet au validateur de transférer une Demande à un autre validateur sans que cette dernière ne soit validée, ni modifiée.

2.6. Demande en relance

Il s'agit d'une fonctionnalité qui permet au validateur de transférer une Demande à un autre validateur sans que cette dernière ne soit validée, ni modifiée.

3. Diagramme d'état

Le diagramme suivant illustre les différents états d'une demande.

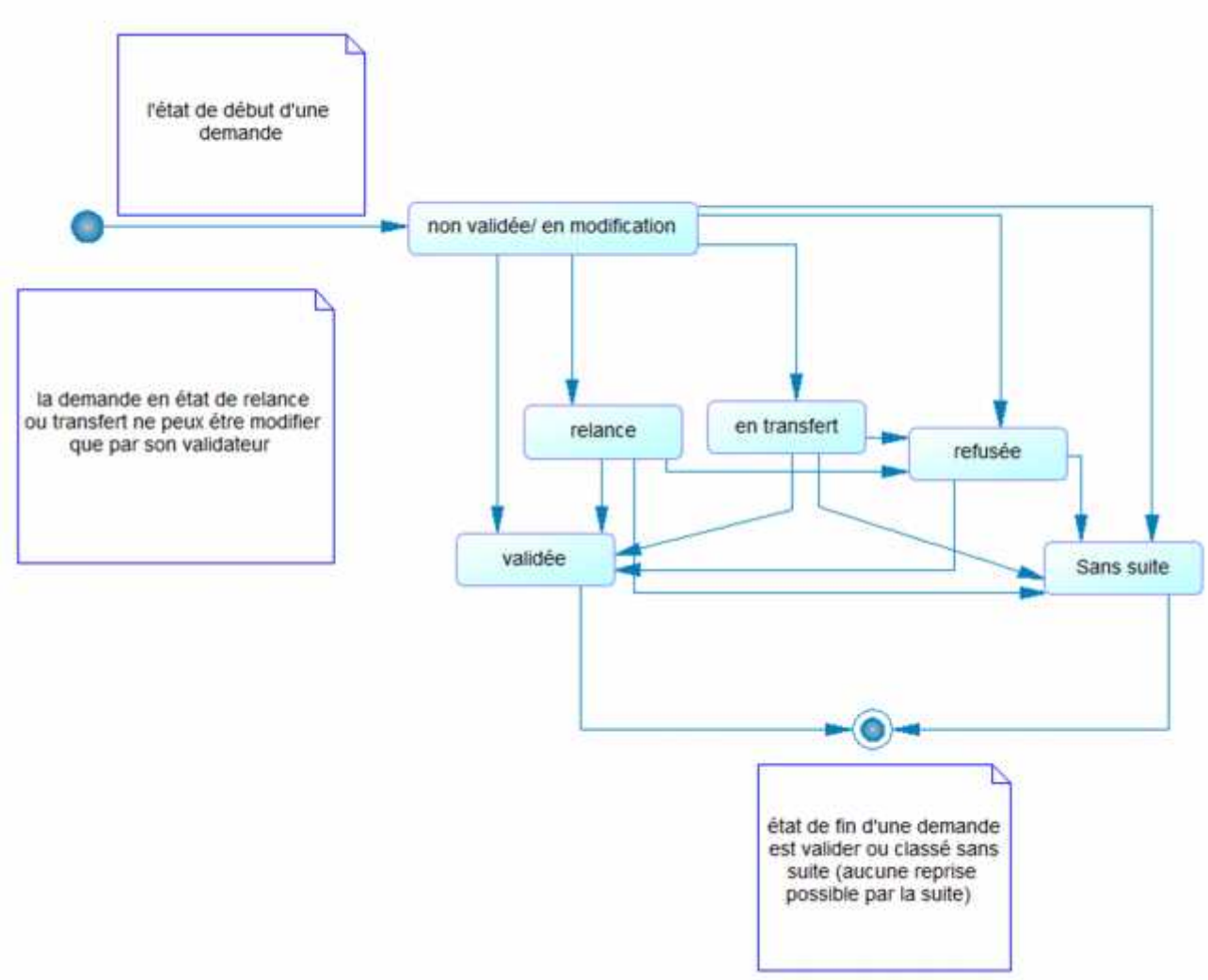


Figure 10. Diagramme d'état de transition d'une demande

4. Diagrammes de séquences

Le diagramme suivant (figure 11) représente la séquence de démarrage avec une authentification machine.

Tous les utilisateurs accèdent à leur machine avec leur Dicid (le login) le système attribue les droits suivant le profil du Dicid.

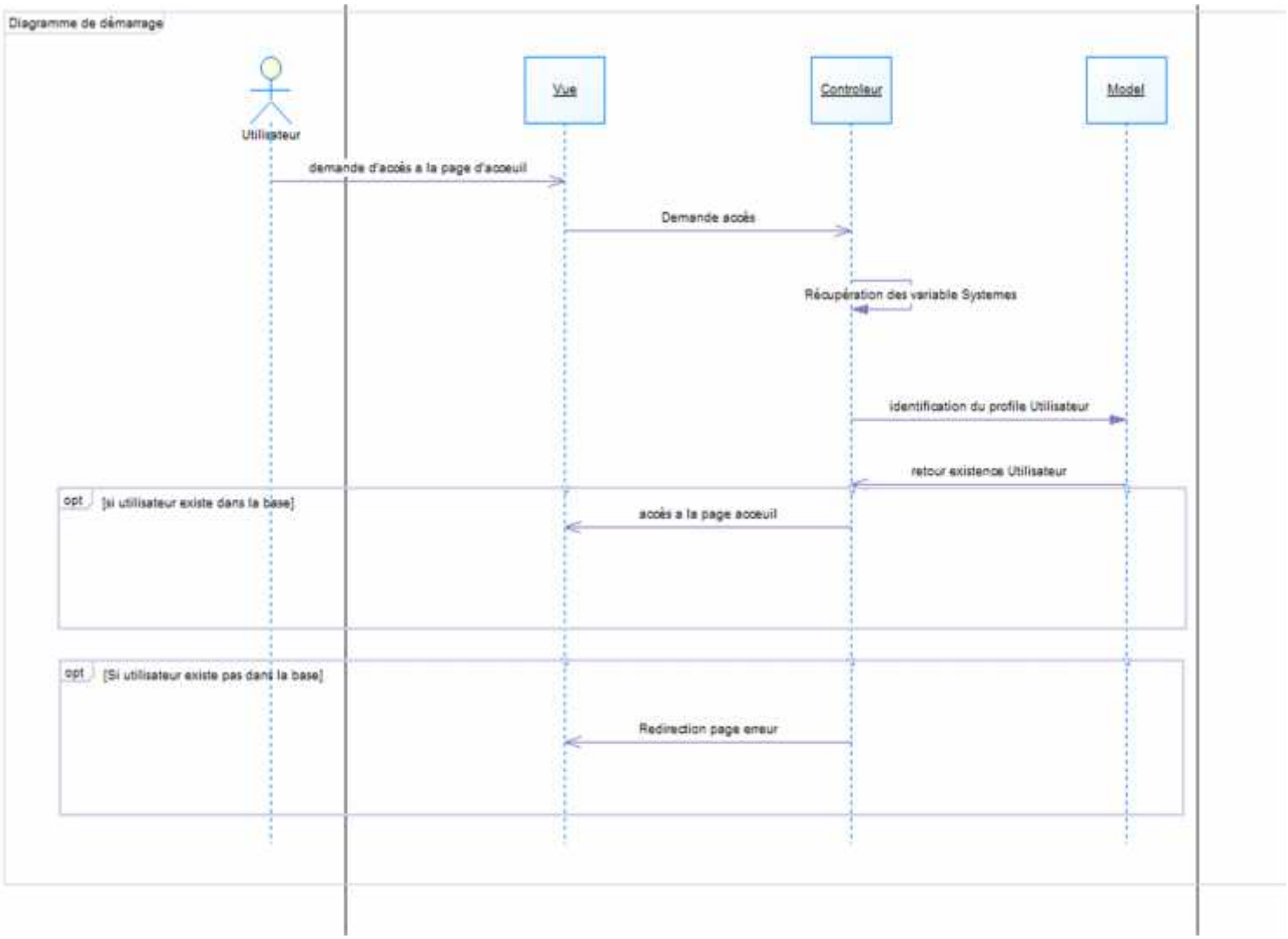


Figure 11. Diagramme de séquence du chargement page accueil

Un validateur peut accéder à une demande soit par un lien reçu par courrier électronique si elle lui a été adressée soit par une recherche via la page de recherche.

La figure 12 représente le diagramme de séquence du traitement d'une demande de la part du validateur.

Le système affiche la demande en état modifiable si elle n'est pas en relance pour un autre validateur ou sans suite ou valider, dans ce cas le validateur modifie l'état de la demande, saisie une justification du changement et valide le changement.

Le système doit enregistrer les modifications et ajouter à la table historique l'état précédent.

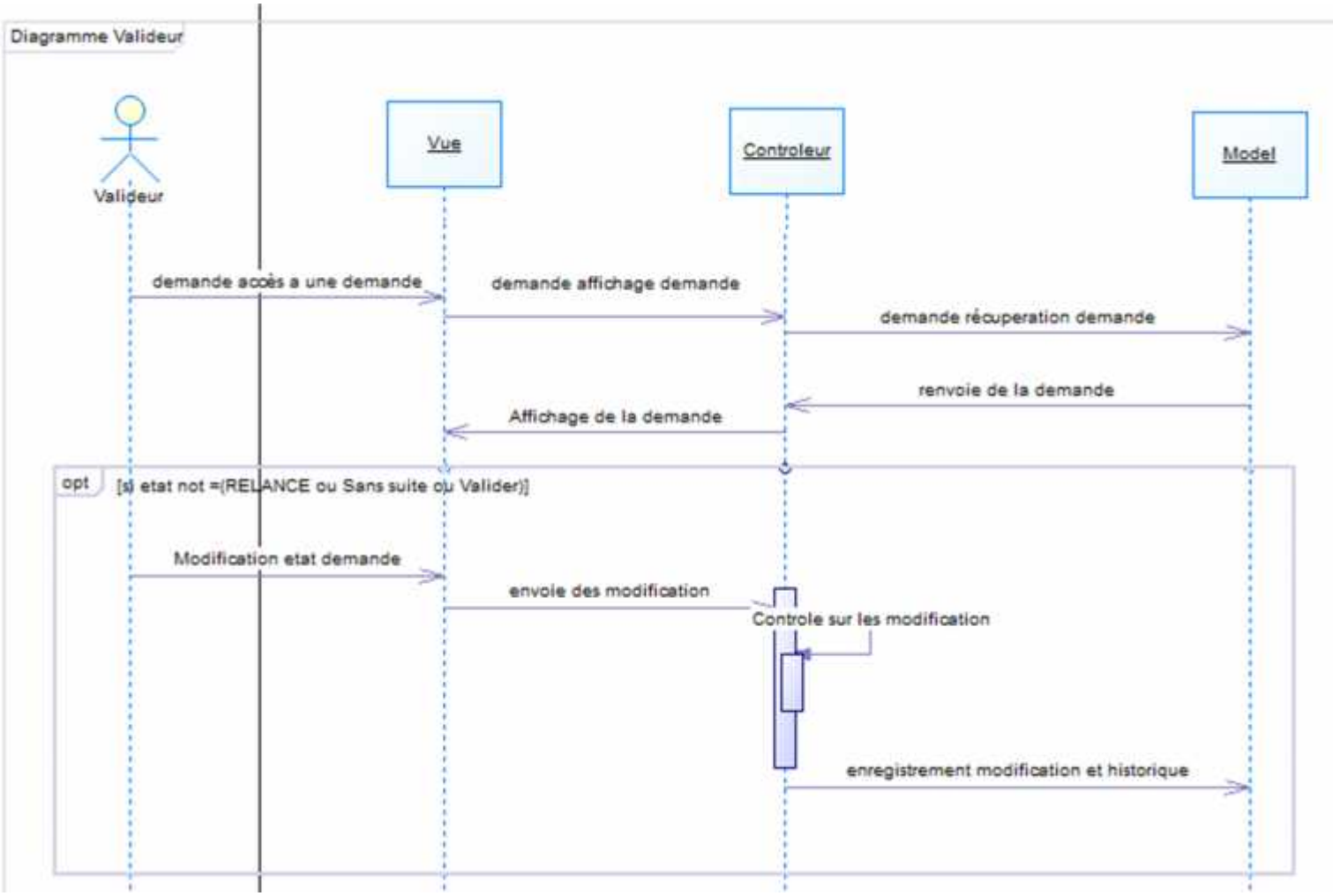


Figure 12. Diagramme de séquence du traitement d'une demande

Le diagramme de séquence figure 13 représente la séquence de traitement pour la saisie d'une demande :

- Demande de saisie formulaire.
- Récupération des champs des différentes tables de la base de données.
- Traitement sur les données pour l'affichage.
- Affichage du formulaire.
- Contrôle sur la saisie de la demande.
- Sauvegarde de la demande dans la base de données.

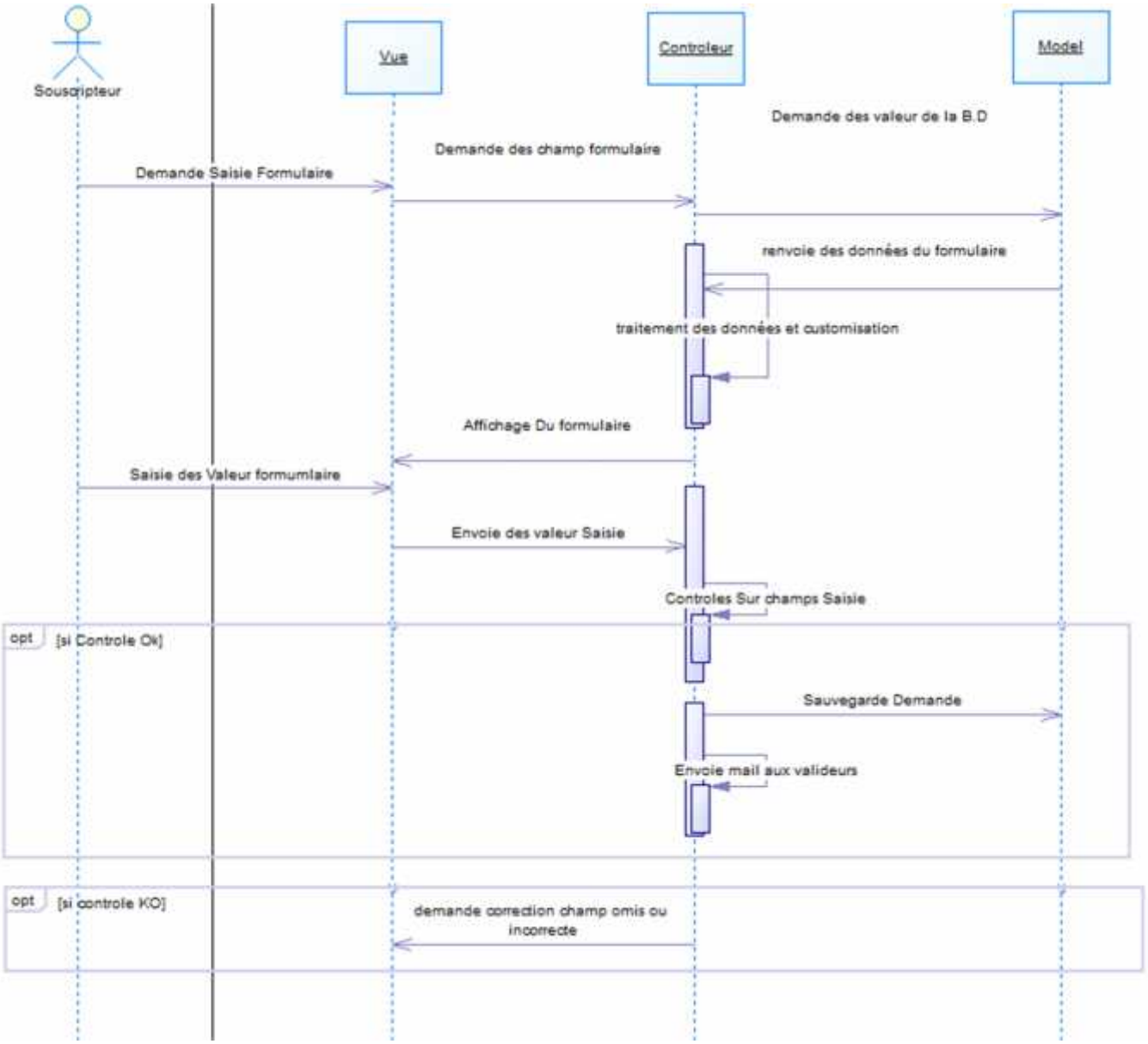


Figure 13. Diagramme de séquence de saisie demande

5. Architecture physique

L'architecture physique assure que les différents composants logiciels, déployés à différents endroits du système informatique, collaborent afin de fournir les fonctionnalités attendues.

Nous enchaînons maintenant par la description de l'architecture physique sur laquelle s'exécutent les différents composants logiciels des modules cibles de notre projet, ainsi que la façon par laquelle ils communiquent entre eux.

L'architecture la mieux adaptée à cette application est une architecture client/serveur 3-tiers, vu qu'elle sépare les trois couches logicielles de l'application.

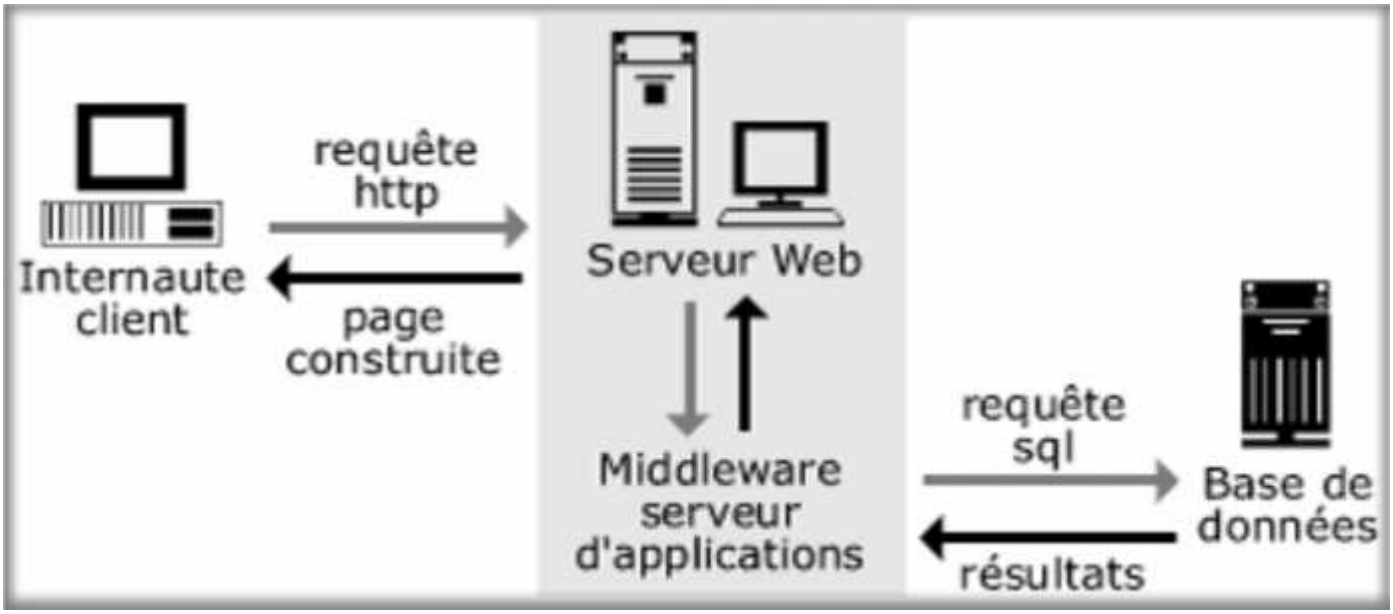


Figure 14. Architecture physique du projet

Le choix est justifié par plusieurs raisons :

Les utilisateurs accèdent à l'application à distance à travers des requêtes « http ». De plus, le besoin d'assurer une grande souplesse, une sécurité flexible, une centralisation des traitements, et une meilleure performance, implique un partage des tâches entre les différents serveurs, c'est-à-dire que les applications sont délocalisées au niveau du serveur, et chaque serveur est spécialisé dans une tâche bien précise.

La solution à réaliser va intégrer :

- **Un client** : équipé d'une interface utilisateur (un navigateur web) chargée de la présentation.
- **Un serveur d'application (appelé également middleware)**: qui fournit la ressource, tout en faisant appel à un autre serveur.
- **Un serveur de base de données** : fournit au serveur d'application les données requises pour répondre au client.

6. Architecture logique

Toujours dans l'optique de séparer les données, les traitements et la présentation, On a opté pour l'architecture logique MVC2. Ainsi, l'application se retrouve décomposée en cinq couches essentielles :

- View : couche spécifique aux interfaces de l'application et qui comprend principalement les pages et formes web (jsf).

- Controller : le contrôleur traite les différentes requêtes d'utilisateurs faisant appel aux divers composants de l'application. Chaque requête atteint d'abord la classe contrôleur frontal. Cette dernière inspecte les demandes entrantes et les achemine à la classe appropriée.
- Dans l'architecture qu'on a adopté (MVC2), ce composant figure sous la notation [Dispatcher Servlet] mais encapsulé par le Framework utilisé.
- Model : représente le retour préparé par le contrôleur pour servir une vue donnée, cette dernière est définie à l'aide des annotations ou dans un fichier de mapping.
- Métier : implémente les algorithmes « métier » de l'application. Cette couche est indépendante de toute forme d'interface directe avec l'utilisateur. Elle est chargée d'exécuter les fonctionnalités métiers de l'application. Constituée essentiellement des entités images extraites à partir de la base de données.
- DAO : la couche d'accès aux données, s'occupe de l'implémentation des opérations CRUD (Create Retrieve Update Delete) sur les objets persistants.

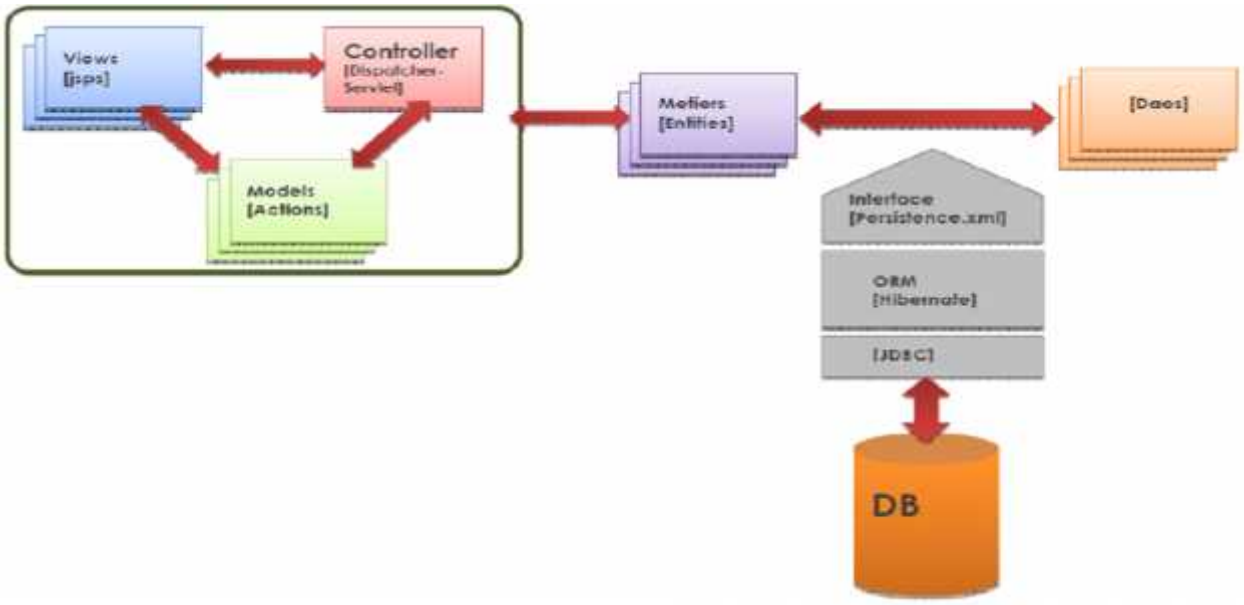


Figure 15. L'architecture en couche de notre application

7. Conclusion :

La capture des besoins techniques est une étape de prise en compte des contraintes techniques et logicielles. Elle doit être suffisamment détaillée pour permettre d'aborder la conception générique du système au cours de ce chapitre nous avant abordé les spécifications techniques de notre projet.

Chapitre 4 : CONCEPTION

1. Avant-propos

Au cours de ce chapitre nous allons aborder la conception de notre application à travers les diffèrent classes qui représente notre modèle et la structuration en packages du programme général.

2. Identification des classes candidates

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse.

Le diagramme de classe est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets.

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

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

Et à partir de la description détaillée des cas d'utilisation, on a créé le diagramme de classe suivant :

2.1. Diagramme de classe

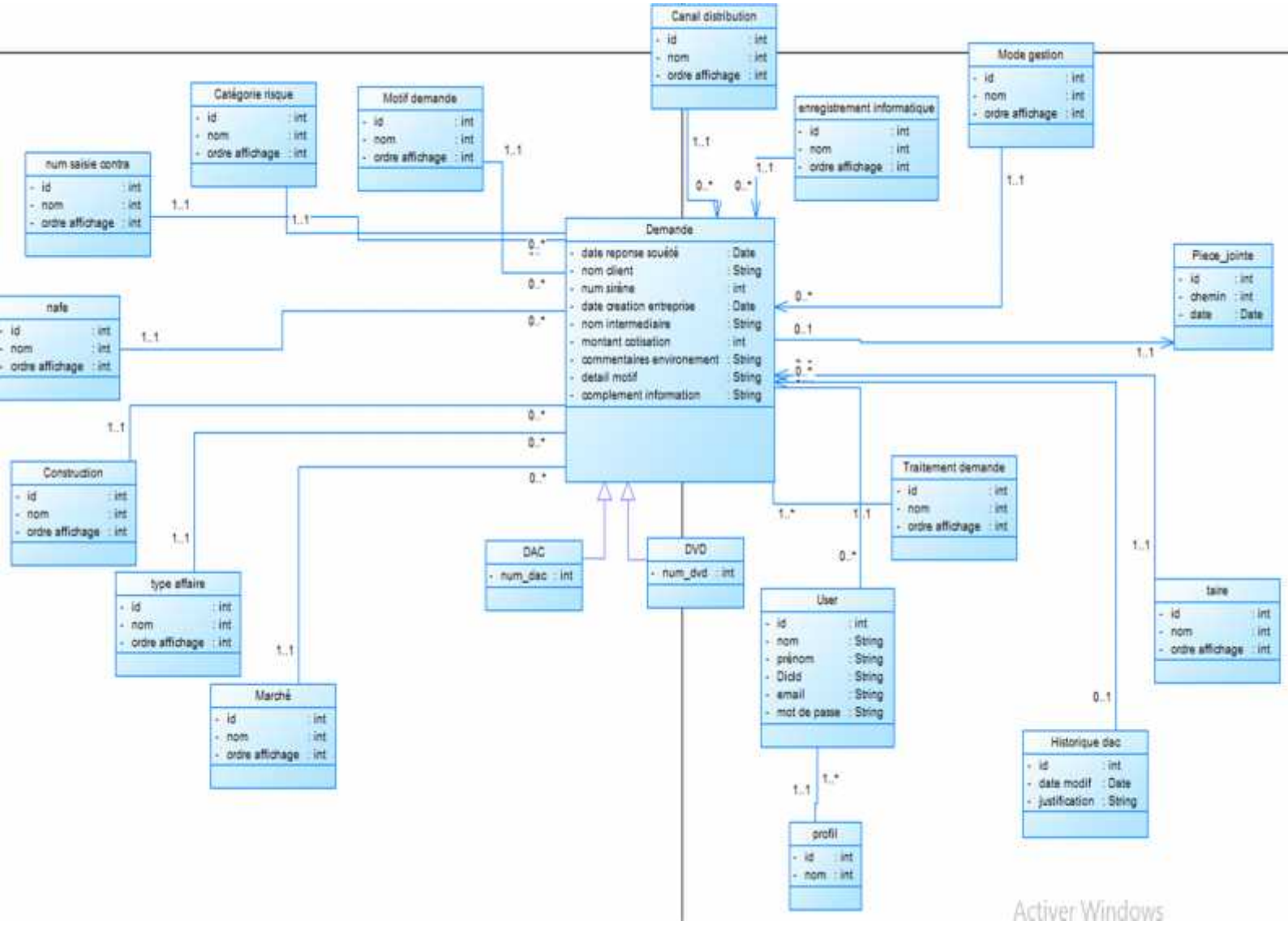


Figure 16. Diagramme de classe du Système

3. Description des packages et structure programme

La structuration de notre programme a été réalisé suivant les packages de la figure 17 à savoir :

- Package Model
- Package couche DAO
- Package Services
- Package Utile
- Package web

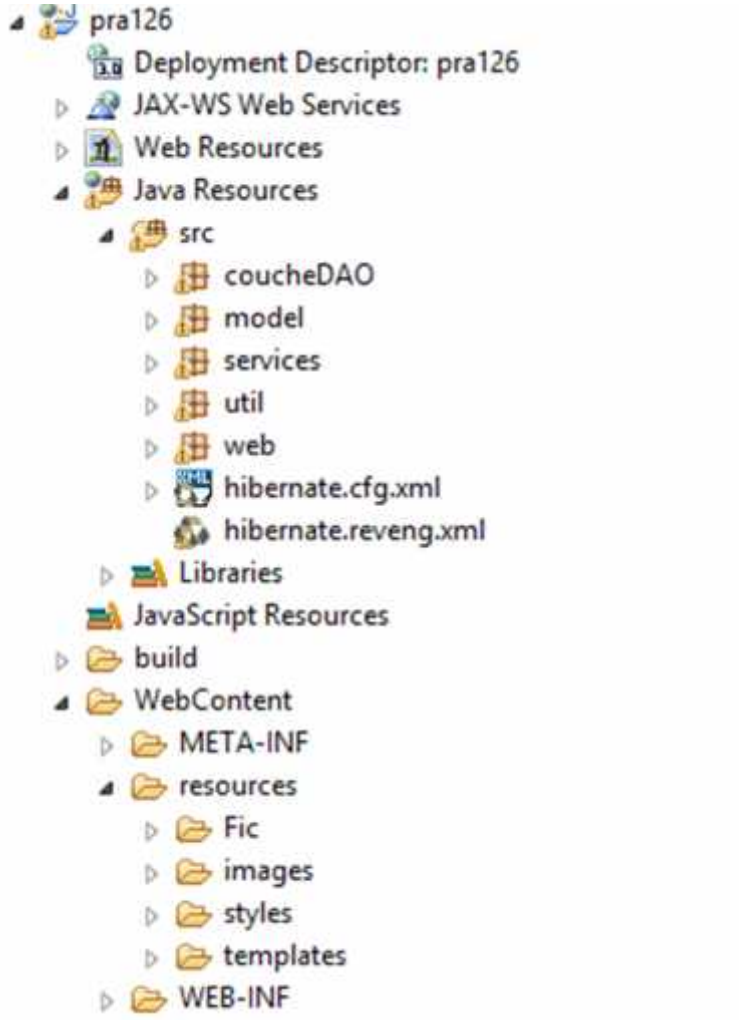


Figure 17 structure en package du système

- Le package Model regroupe les classes de la figure 16

- Le package coucheDAO regroupe l'ensemble des interactions avec notre base de donnée avec des requête HQL.

- Le package services regroupe toutes les fonctionnalités du système à savoir l'envoi des courriers électronique avec java mail, les class de lecture et écriture de fichier pour les modèles de courrier et une invocation des Services à travers L'interface du package DAO.

- Le package Utile qui est propre à la gestion des transactions Hibernante.

- Le package Web Regroupe L'ensemble des Managed beans pour les transactions Web.

Le répertoire ressources contient :

- Un répertoire Fic pour les modèles de mail à envoyer.
- Un répertoire images.
- Un répertoire templates pour les Templates de structuration de notre application.
- Un Répertoire styles pour notre partie en CSS.

4. Conclusion

Au cours de ce chapitre, on a entamé la partie conception qui constitue l'aboutissement du projet et la concrétisation des phases d'analyse et de conception, où on a présenté la structure du projet en package détailler le rôle de chaque package.

Chapitre 5 : mise en œuvre

1. Avant-propos

Ce chapitre est essentiellement consacré à la présentation des différents outils utilisés lors de la mise en œuvre de cette application et des principales interfaces du système sous forme de capture d'écrans. Chaque écran est précédé d'un commentaire pour décrire son fonctionnement.

2. Outils de développement

2.1 Environnement de développement Eclipse

Eclipse est un IDE, *Integrated Development Environment* (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Développé par IBM, ce logiciel est gratuit et disponible pour la plupart des systèmes d'exploitation.
Son intérêt se situe dans la facilitation de la mise en place de différents Framework.

2.2 Serveur web et d'application : Apache Tomcat

Je me suis servi de Tomcat comme serveur d'application de notre projet. Comme il inclut un serveur http interne, il est aussi considéré comme serveur http. Tomcat est souvent utilisé en association avec un autre serveur web, en général Apache.

2.3 WampServer :

WampServer est une plate-forme de développement Web sous Windows. Elle permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Elle possède également PHPMyAdmin pour gérer plus facilement les bases de données.

Contrairement aux autres solutions, WampServer permet de reproduire fidèlement le serveur de production. Une fois la base installée, on a la possibilité d'ajouter autant de versions d'Apache, MySQL et PHP que nous le souhaitons.

2.4 Entreprise Architect :

C'est un outil d'analyse et de création UML, couvrant le développement de logiciels du rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien. Cet outil graphique basé sur Windows, peut être utilisé par plusieurs personnes et conçu pour aider à construire des logiciels faciles à mettre à jour. Il comprend un outil de production de documentation souple et de haute qualité.

3. Framework de développement :

Framework de persistance : Hibernate

Hibernate vu qu'il est un Framework open source qui me permettra de réaliser, d'une manière automatique et en utilisant le serveur d'application Tomcat, le mapping entre les objets et les données stockées dans une base de données relationnelle. Son utilisation optimisera le temps réservé par le développement. C'est aussi un Framework qui permet facilement la migration vers une autre BD.

Framework java mail :

Java Mailest une API qui permet la gestion des courriers électroniques (e-mail) dans une application écrite en Java (application cliente, applet, servlet, EJB,..). Son but est d'être facile à utiliser, de fournir une souplesse qui permet de la faire évoluer et de rester le plus indépendant possible des protocoles utilisés.

Framework jsf :

Java Server Faces (abrégé en JSF) est un Framework Java, pour le développement d'applications Web.

À l'inverse des autres Frameworks MVC traditionnels à base d'actions, JSF est basé sur la notion de composants, comparable à celle de Swing ou SWT, où l'état d'un composant est enregistré lors du rendu de la page, pour être ensuite restauré au retour de la requête.

4. Interfaces Graphiques

4.1. Interface d'accueil

La page d'accueil reste inchangé l'application sera intégrer à un module général du nom A126 qui vas contenir les nouveaux liens de notre application comme présenté dans la figure 18.

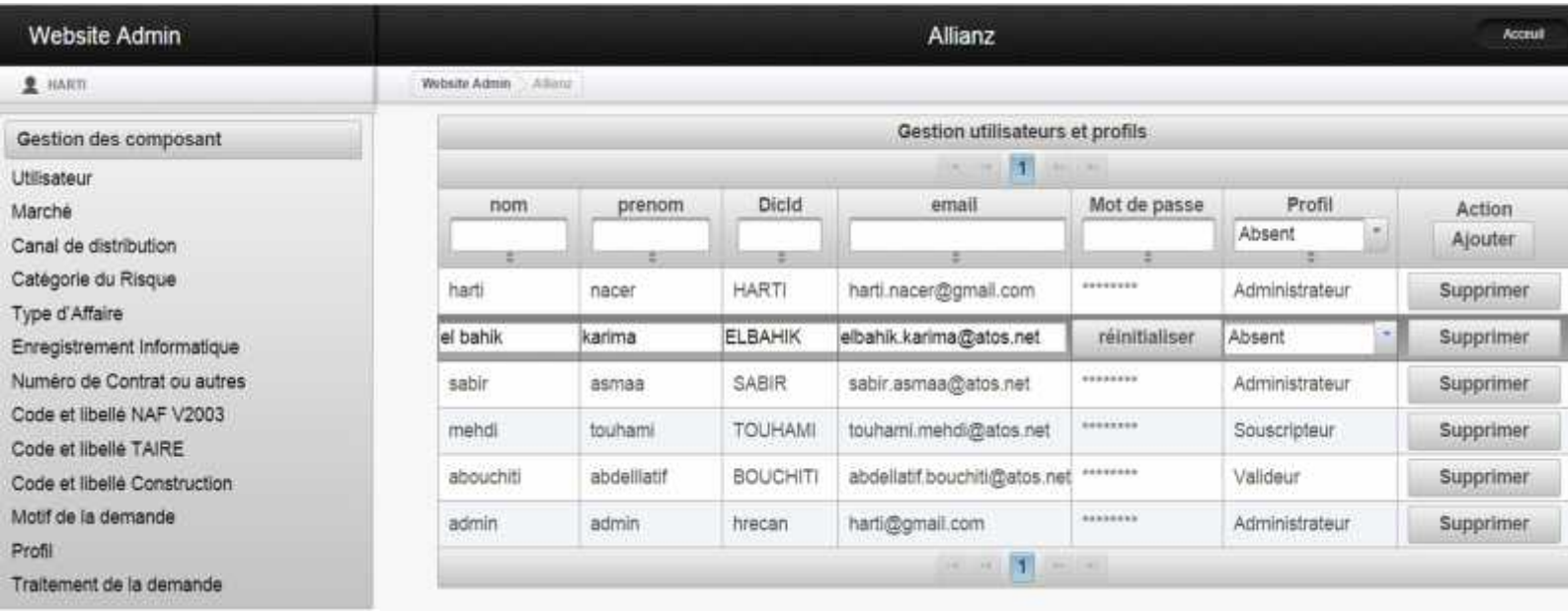
4.2. Interfaces de gestion des composants dynamique

Fenêtre de gestion des modèles de courriers électroniques

4.3. Interface gestion des Utilisateurs

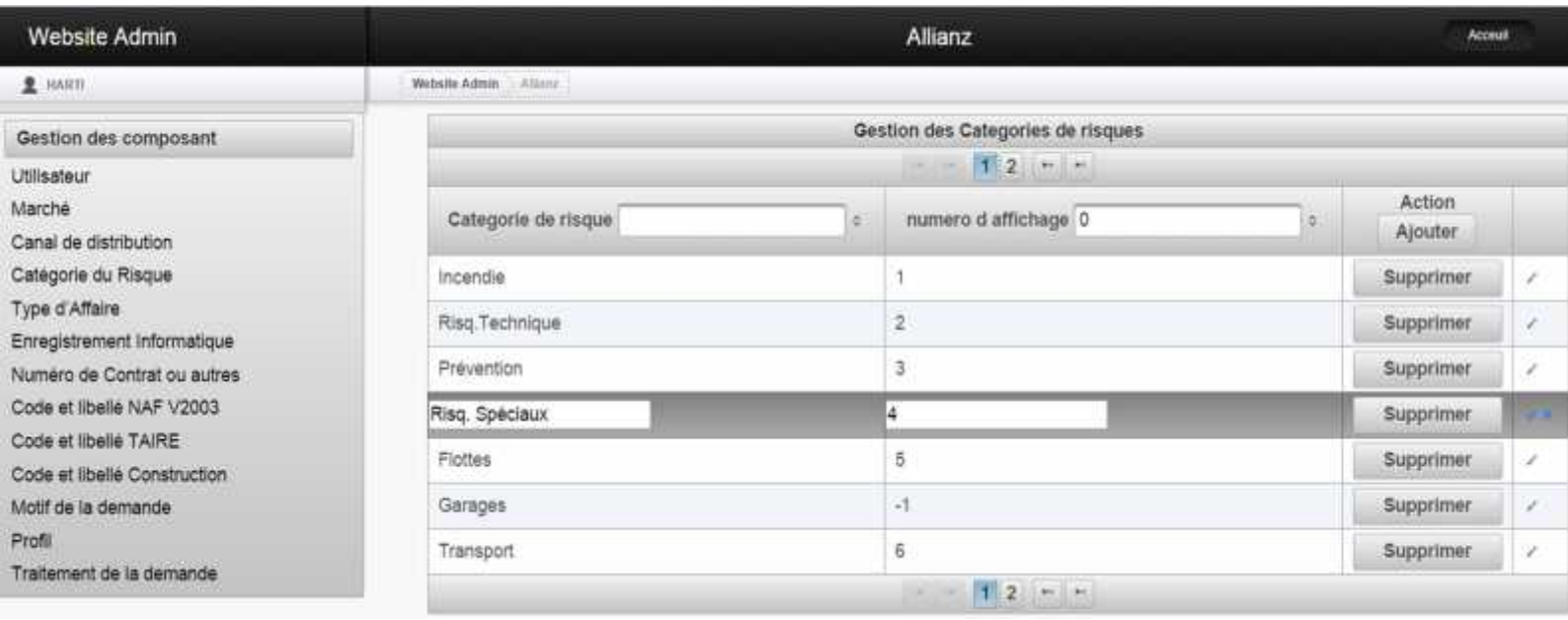
Interface de gestion des utilisateurs avec fonction ajout, suppression, Modification.

Le mot de passe ne peut pas être affiché ou modifier directement, il est généré et envoyer par email.



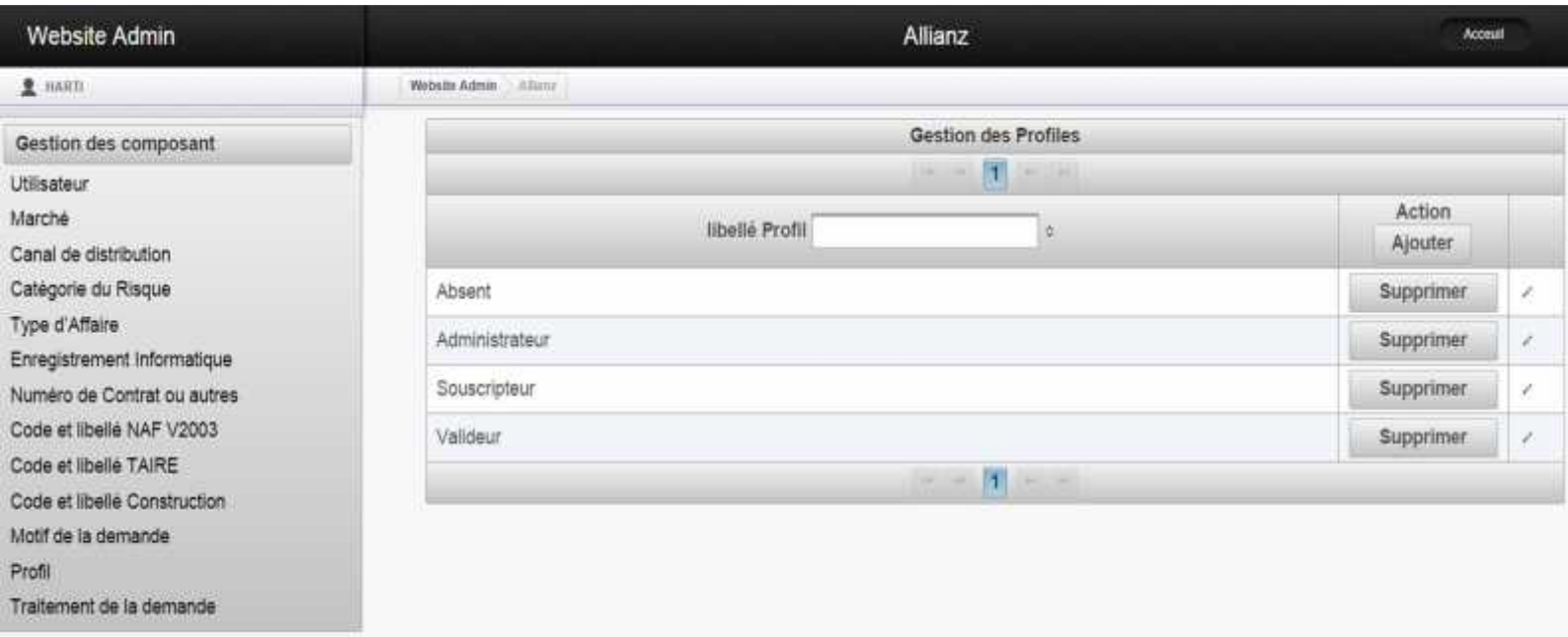
4.4. Interface de gestion des catégories de risques

Interface de gestion des catégories de risque avec les fonction ajout, suppression, Modification.



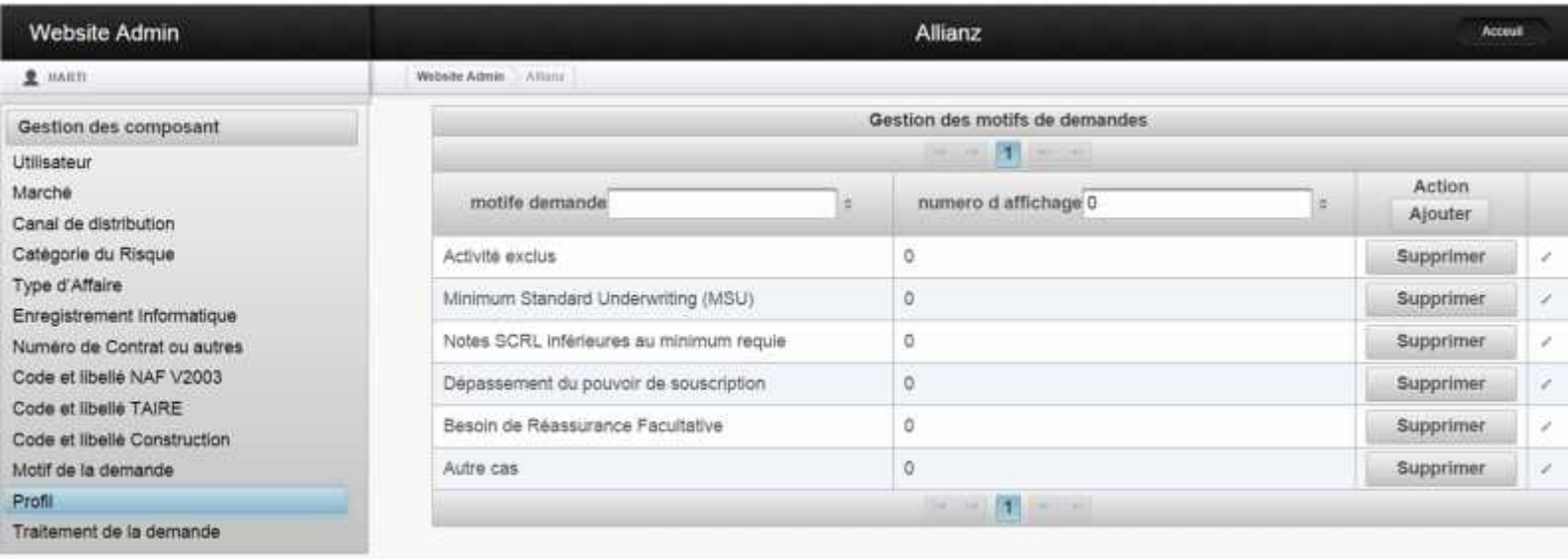
4.5. Interface de gestion des profiles

Interface de gestion des profils avec fonction ajout, suppression, Modification.



4.6. Interface de gestion des motifs de demande

Interface de gestion des motifs de demande avec fonction ajout, suppression, Modification.

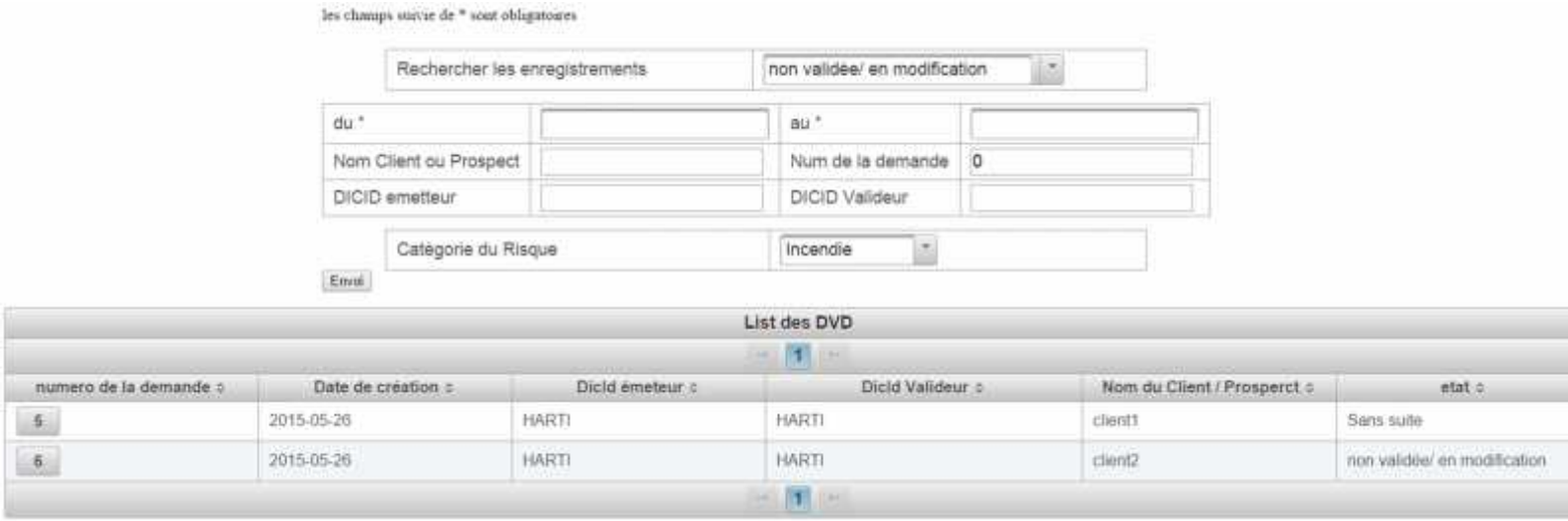


4.7. Interface de gestion des Types affaires

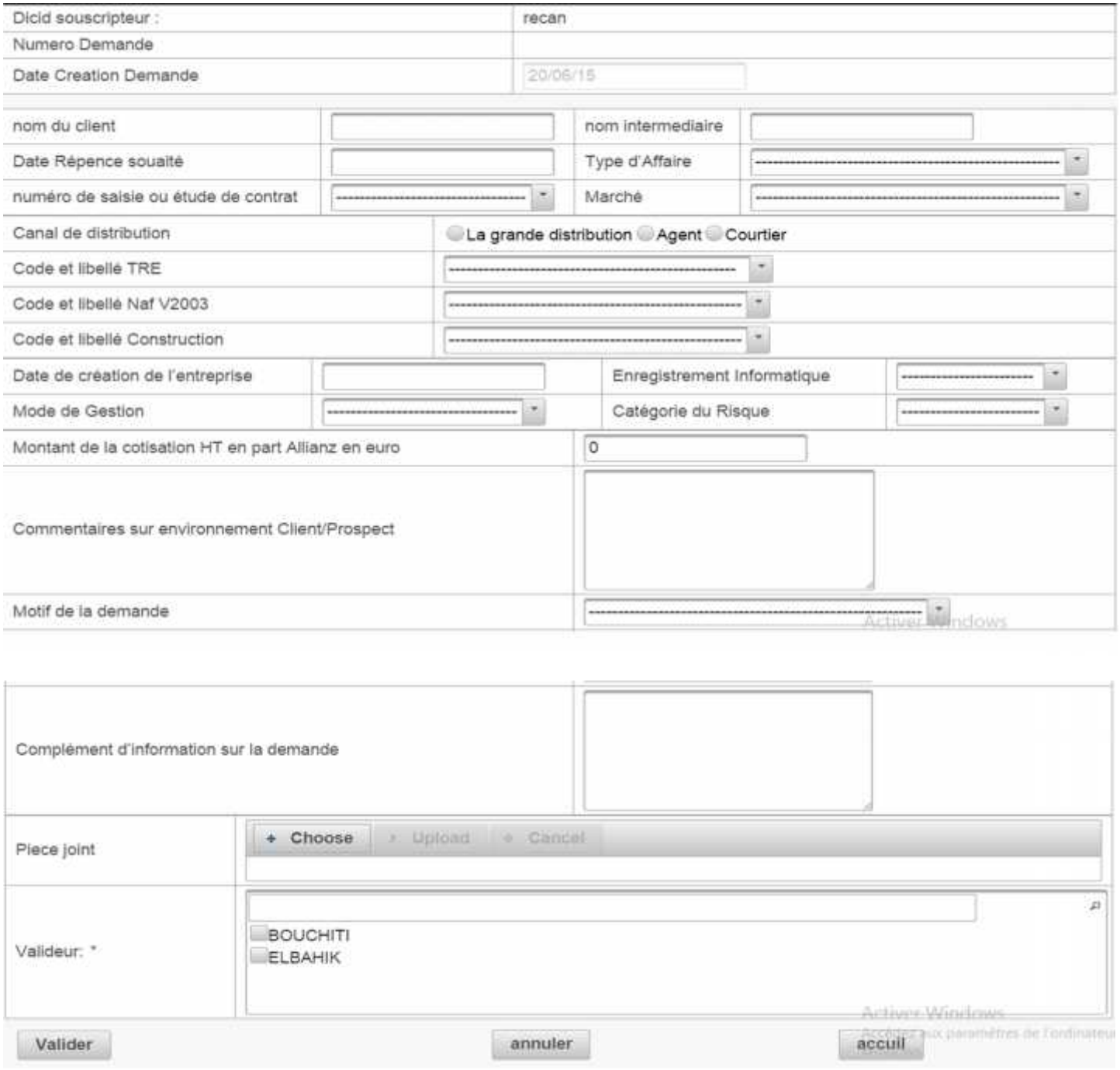
4.8. Interface de consultation des demandes

Interface de consultation et recherche des demandes avec comme critères :

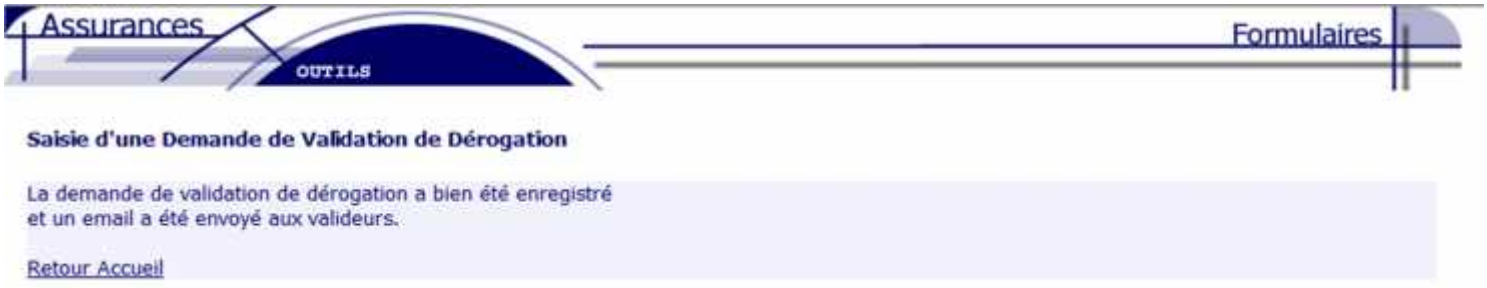
Date début, Date fin, nom client, numéro de la demande, Dicid validateur et souscripteur.



4.9. Interface de saisie d'une demande



4.10. Interface de redirection de la saisie de demande



4.11. Interfaces de consultation demande

Cette interface affiche une demande avec l'historique des traitement de la demande.

Demande numero : 5		non validée/ en modification	
nom du client	client1	nom intermediaire	nacer
Date Répence souaité	31-00-2015	Type d'Affaire :	Agriculture, foresterie, pêche et chas
numéro de saisie ou étude de contrat	DCA24100	Marché	marchés de services
Canal de distribution	La grande distribution	Code et libellé TRE	extraction et préparation de minéraux
Code et libellé Naf V2003	culture de céréales	Code et libellé Construction	BCAL-1639
Date de création de l'entreprise	16-00-1988	Enregistrement informatique	valide
Mode de Gestion	100%	Catégorie du Risque	Incendie
Montant de la cotisation HT en part Allianz en euro	19990	Commentaires sur environnement Client/Prospect	
Motif de la demande	Activité exclus	DEtail Motif	
Complément d'information sur la demande		Piece joint	aucune piece joint disponible

Historique de la Demande

Date Modification	Valideur	Justification	etat de la demande
18/05/15	HARTI	manque d'information justificative	refusée
21/05/15	SABIR		Sans suite

4.12. Interface du choix de validation demande

Cette interface figure sur la page de consultation demande elle est afficher quand la demande peux etre modifier et quand elle est consulter par un validateur.

5. Conclusion

Dans ce chapitre nous avons abordé les principaux outils utilisés pour réaliser ce projet ainsi que les interfaces principales de notre Système.

Conclusion général

L’objectif de mon projet de fin d’études était la conception et la réalisation d'un système d'information de gestion des demandes de dérogation et d’action après Contrôle sur des contrats d’assurance.

Le point de départ de la réalisation de ce projet était une collecte des informations nécessaires pour présenter un aperçu sur la problématique.

Par la suite, je me suis intéressée à la spécification et à l’analyse des besoins, ce qui m'a permis de distinguer les différents acteurs interagissant avec le système visé ainsi que les différentes fonctionnalités qu'il va offrir.

Le dernier volet de mon projet était la partie réalisation qui a été élaborée pour la mise en œuvre des itérations les plus significatives.

Comme tout projet, j'ai rencontré quelques problèmes durant la réalisation; tel que le manque d'information et le suivi.

L’apport de ce travail a été d’une importance très considérable, en effet, il m’a permis de suivre une méthodologie de travail bien étudiée, d’approfondir mes connaissances dans le monde de développement des systèmes d'information et plus précisément les frameworks de JAVA EE, il m'a appris également la nécessité d’entretenir de bonnes relations avec ses collègues afin de profiter d’une bonne ambiance de travail.

L’application que j’ai développée pourrait être enrichie par d’autres fonctionnalités avancées et améliorations peuvent être envisagées pour l’enrichir, tel que:

- Des Interfaces plus ergonomiques.
- Une meilleure gestion des erreurs.
- Une meilleure structuration du programme pour de future évolution de notre system.

Bibliographie

1. Les ouvrages

Jean-Michel DOUDOUX. Développons en Java Disponible sur :
http://www.jmdoudoux.fr/accueil_java.htm#dej

2. Les sites-web

http://java.developpez.com/faq/jsf/
Une FAQ sur l'utilisation des JSF. Très utile lorsque l'on a un problème souvent rencontré par les développeurs car on y trouve une solution rapidement.
http://www.primefaces.org/showcase/
Site qui montre des exemples d'utilisation de Primefaces.
http://www.e-bancel.com/Processus_2TUP.php
Site dans lequel, j'ai appris comment implémenter la notion des incréments.
http://www.jmdoudoux.fr/java/dej/chap-jsf.htm
Site qui contient des exemples d'utilisation des composants JSF bien détaillés.

3. Forums

http://www.developpez.net/forums/f300/java/developpement-web-java/frameworks/jsf/
http://stackoverflow.com/questions/2206911/best-way-for-user-authentication-on-javaee-6-using-jsf-2-0
http://stackoverflow.com/questions/8367698/jpa-and-hibernate-mapping-with-mysql-field-describtion-doesnt-have-a-default
http://stackoverflow.com/questions/8699195/cannot-be-cast-to-java-lang-comparable