

**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 de gestion des projets
aéronautiques et calcul des indicateurs**



Lieu de stage :Technopolis Rabat-Salé

**Réalisé par : MOUHIB Ibtihal
Soutenu le 23/06/2012**

Encadré par :

Pr Mohammed ChaoukiABOUNAIMA
Pr Ilham CHAKER
MrDRISSI El BouzidiAnouar (Zodiac)

Devant le jury composé de :

Pr. M.OUZARF
Pr. M.C. ABOUNAIMA
Pr.I.CHAKER
Pr. A.LACHKAR
Mr DRISSI El Bouzidi Anouar (Zodiac)

ANNÉE UNIVERSITAIRE 2011-2012

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

نَشَأُ أَذْكَرَ مَنْكُمْ عَامِلِ عَمَلٍ أَضْيَعُ لَانِّيَارِ بِيْهُمُ لَهُمْ فَاَسْتَجَابَ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿قُلْ هُوَ اللَّهُ أَحَدٌ ۝ (١) اللَّهُ الصَّمَدُ ۝ (٢) لَمْ يَكُنْ لَهُ كُفُوًا أَحَدٌ ۝ (٣)﴾

www.facebook.com/w.Bsher

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿قُلْ أَعُوذُ بِرَبِّ الْفَلَقِ ۝ (١) مِنْ شَرِّ مَا خَلَقَ ۝ (٢) وَمِنْ شَرِّ غَاسِقٍ إِذَا وَقَبَ ۝ (٣) وَمِنْ شَرِّ النَّفَّاثَاتِ فِي الْعُقَدِ ۝ (٤) وَمِنْ شَرِّ حَاسِدٍ إِذَا حَسَدَ ۝ (٥)﴾

www.facebook.com/w.Bsher

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿قُلْ أَعُوذُ بِرَبِّ النَّاسِ ۝ (١) مَلِكِ النَّاسِ ۝ (٢) إِلَهِ النَّاسِ ۝ (٣) مِنْ شَرِّ الْوَسْوَاسِ الْخَنَّاسِ ۝ (٤) الَّذِي يُوَسْوِسُ فِي صُدُورِ النَّاسِ ۝ (٥) مِنَ الْجِنَّةِ وَالنَّاسِ ۝ (٦)﴾

صَدَقَ اللَّهُ الْعَظِيمُ

Dédicaces

*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 très chers parents : Ma mère et mon père
La source de ma force qui m'a permis d'endurer toutes les difficultés, de surmonter les défis et de surpasser soi dans le but de réaliser mes objectifs. Cette réussite c'est la vôtre, c'est vous qui méritent d'être félicités. J'espère être à la hauteur de l'image que vous m'avez de moi. Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.*

*A la mémoire de mon grandpère
Je sais que ma réussite est une fierté pour lui. Puisse Dieu l'accueillir dans son infinie Miséricorde.*

A ma chère sœur Hala et mon cher frère Mohammed

Merci pour votre amour.

A mes chères amies : Amal, maria, Hind

A mes chers amis : Yassir, Imad

Je vous aime,

A ma chère colocataire : Kaoutar

Qui m'a donné plein d'amour et de tendresse tout au long de ma période de stage.

A mes chers collègues de Zodiak

Qui m'ont beaucoup encouragé, qui m'ont aidé à reprendre mon souffle

A Kacem BenTaleb

Qui m'a beaucoup aidé dans la partie développement du projet. Merci.

A tous mes autres amis

A tous ceux qui m'aiment...

Remerciement



Le projet que j'ai effectué au sein de la société Zodiac Aerospace Maroc n'aurait pu être mené à bien sans le soutien et l'appui d'un grand nombre de personnes, qui trouvent ici l'expression de mes profondes gratitude.

Je tiens à exprimer mes remerciements à **Monsieur Frédéric ARCIET**, le Directeur Général de la société Zodiac Aerospace Maroc, pour m'avoir permis d'effectuer mon stage de fin d'études durant ces quatre mois au sein de ZAM.

J'adresse mes remerciements à l'ensemble du personnel de Zodiac Aerospace Maroc et plus particulièrement **Monsieur Xavier BAUDASSE**, Responsable du pôle ingénierie de ZAM, **Monsieur Mohamed BENABBOU**, Responsable de l'équipe D7/INTERTECHNIQUE, pour leur accueil sympathique et leur coopération professionnelle tout au long de mon stage.

J'exprime mes remerciements à **Monsieur Khalid SAFI**, Responsable de l'équipe Tests Unitaires et Vérification & Validation qui n'a ménagé ni son temps ni son énergie pour m'aider à élaborer mon projet dans les meilleures conditions. À mon encadrant **Monsieur Anouar DRISSI EL-BOUZAI** pour sa confiance, ses instructions et ses conseils tout au long de la réalisation de ce projet.

Je remercie également mes encadrants de la FSTF **Monsieur Mohammed CHAOUKI ABOUNAIMA** et **Madame Ilham CHAKER** pour leurs précieuses directives et pertinents conseils qu'ils m'ont prodigués.

J'aimerais également remercier tous les membres de jury d'avoir accepté de juger mon travail.

Enfin je remercie toutes les personnes qui m'ont soutenue, d'une façon ou d'une autre, m'éprouvant incessamment leur estime et amabilité, je salue réellement cette très haute bienveillance que vous portez à mon égard et qui me marquera toujours. Que la paix de Dieu soit toujours avec vous!

Résumé

Dans un contexte de mondialisation et de concurrence chaque jour un peu plus sévère, la maîtrise des coûts et des délais est devenue une priorité pour tous les acteurs de l'industrie.

Or, le moins que l'on puisse dire, c'est que ce n'est pas le point fort de l'industrie logicielle, qui est habituée à des dépassements qui seraient jugés inacceptables dans bien d'autres contextes.

Zodiac Aerospace Maroc a ainsi entamé une démarche d'amélioration de la qualité pour assurer à ses clients et ses collaborateurs les mêmes pratiques, le même engagement et la même rigueur.

Mon projet de fin d'étude, réalisé au sein du pôle ingénierie de Zodiac Aerospace Maroc, porte sur la conception et la mise en place d'une application JEE ayant comme objectif d'automatiser le flux de processus de gestion des projets et le calcul des indicateurs.

Le projet consiste à mettre à la disposition des membres de l'équipe, chef d'équipe, développeurs, vérificateurs, ainsi que d'autres supérieurs hiérarchiques un outil flexible et fiable leur permettant de gérer les différents projets logiciels embarqués aéronautiques, de suivre les travaux réalisés et leurs livraisons, de calculer et de visualiser les indicateurs de performance de l'équipe Tests Unitaires et Vérification & Validation.

TABLE DES MATIERES

DEDICACES	3
REMERCIEMENT	4
RESUME	5
LISTE DES FIGURES.....	8
LISTE DES ABREVIATIONS.....	10
LISTE DES TABLEAUX.....	10
INTRODUCTION GENERALE.....	12
PARTIE I : CONTEXTE GENERAL DU PROJET	13
CHAPITRE I : PRESENTATION DE L'ORGANISME D'ACCUEIL	14
I.1.LE GROUPE ZODIAC AEROSPACE.....	15
I.2. ZODIAC AEROSPACE MAROC (ZAM)	19
I.3. PRESENTATION DE L'EQUIPE DE TRAVAIL.....	21
I.4. METHODOLOGIE ET CYCLE DE VIE LOGICIEL DE L'EQUIPE D7	21
CHAPITRE II : PRESENTATION GENERALE DU SUJET.....	24
II.1.LIMITES DES METHODOLOGIES ACTUELLES.....	25
II.2. PRESENTATION DU SUJET	25
II.3. COMPOSANTES DE LA SOLUTION	25
II.4. CONDUITE DE PROJET.....	25
PARTIE II : ETUDE CONCEPTUELLE.....	30
CHAPITRE I : ETUDE PRELIMINAIRE ET CAPTURE DES BESOINS FONCTIONNELS.....	31
I.ETUDE PRELIMINAIRE.....	32
I.1. Besoins fonctionnels	32
II. Capture des besoins fonctionnels	34
II.1. Identification et description des Cas d'utilisations	35
II.2. Identification des classes candidates.....	46
CHAPITRE II : CAPTURE DES BESOINS TECHNIQUES ET ANALYSE	47
II.1. CAPTURE DES BESOINS TECHNIQUES	48
II.1.1 Organisation du modèle de spécification logicielle	48
II.1.2. Définition des concepts techniques.....	49
II.2.ANALYSE	50
II.2.1.Découpage en catégories	50
II.2.2. Développement du modèle statique	54
II.2.3. Développement du modèle Dynamique.....	55
CHAPITRE III : CONCEPTION GENERIQUE, PRELIMINAIRE ET DETAILLEE.....	61
III.1.CONCEPTION GENERIQUE.....	62
III.1.1. Classes et frameworks techniques	62
III.1.2. Organisation des frameworks techniques	63
III.1.3 Développement du prototype technique	63
III.2.CONCEPTION PRELIMINAIRE	64
III.2.1. Développement du modèle de déploiement.....	64
III.2.2 Développement du modèle d'exploitation.....	65
III.2.3. Énumération des interfaces utilisateur.....	66

III.3 CONCEPTION DETAILLEE.....	69
III.3.1 Le micro-processus de conception logique.....	69
IV.3.2. Description du modèle de classes.....	70
IV.3.3. Conception des couches de l'application	71
PARTIE III : MISE EN ŒUVRE.....	74
CHAPITRE I : OUTILS ET TECHNOLOGIES	75
CHAPITRE II : PROTOTYPAGES DES INTERFACES	79
CONCLUSION ET PERSPECTIVES	97
BIBLIOGRAPHIE	0
WEBOGRAPHIE.....	0
ANNEXES	1

Liste des figures

Figure 1 : Organisation en branches du groupe Zodiac Aerospace	15
Figure 2: Jauge de réservoir	17
Figure 3 : Indicateur à aiguilles.....	17
Figure 4: Calculateur	18
Figure 5: Organigramme de Zodiac Aerospace Maroc (ZAM).....	20
Figure 6 : Organigramme du pôle ingénierie du ZAM.....	20
Figure 7 : Organigramme de l'équipe D7.....	21
Figure 8 : Cycle en V - INTERTECHNIQUE.....	22
Figure 9 :Stratégie de Test - INTERTECHNIQUE.....	23
Figure 10: Planning du projet.....	26
Figure 11 : Le processus de développement 2TUP	27
Figure 12: Situation de l'étude préliminaire dans 2TUP	32
Figure 13: Situation de l'étude préliminaire dans 2TUP	34
Figure 14: Diagramme de cas d'utilisation « Rôle »	35
Figure 15: Diagramme de cas d'utilisation « Poste ».....	36
Figure 16: Diagramme de cas d'utilisation « Intervenant »	42
Figure 17: Diagramme de cas d'utilisation « Activité ».....	38
Figure 18: Diagramme de cas d'utilisation « Type d'activité »	39
Figure 19: Diagramme de cas d'utilisation « Branche »	40
Figure 20: Diagramme de cas d'utilisation « Projet »	41
Figure 21: Diagramme de cas d'utilisation « Package »	42
Diagramme de cas d'utilisation « Livraison ».....	43
Figure 23: Diagramme de cas d'utilisation « Congé »	44
Figure 24: Diagramme de cas d'utilisation « Compte ».....	45
Figure 25: Diagramme des classes candidates	46
Figure 26: Situation de la capture des besoins techniques dans 2TUP	48
Figure 27: Architecture en couches.....	48
Figure 28: diagramme de packages - Organisation du modèle de spécification logicielle	49
Figure 29: Situation du découpage en catégories dans 2TUP	50
Figure 30 : Diagramme de packages	50
Figure 31 : Diagramme de classes de la Catégorie Package	51
Figure 32:Diagramme de classes de la Catégorie Activité.....	52
Figure 33:Diagramme de classes de la Catégorie Branche	52
Figure 34:Diagramme de classes de la Catégorie projet	53
Figure 35 :Diagramme de classes de la Catégorie intervenant.....	53
Figure 36:Diagramme de classes de la Catégorie indicateur.....	54
Figure 37: Situation du modèle statique dans 2TUP	54
Figure 38:Diagramme de classe avec les opérations et les attributs.....	55
Figure 39: Situation du développement du modèle dynamique dans 2TUP.....	56
Figure 40: Diagramme de séquence de la partie authentification.....	56
Figure 41: Diagramme de séquence de la partie consultation pour un administrateur.....	57
Figure 42: Diagramme de séquence de la partie consultation pour un utilisateur.....	57

Figure 43: Diagramme de séquence de la partie ajout objet.....	58
Figure 44: Diagramme de séquence de la partie consultation intervenant	59
Figure 45: Diagramme de séquence de la partie consultation package	60
Figure 46: Situation de la conception générique dans 2TUP	62
Figure 47: Situation de la conception générique dans 2TUP	64
Figure 48: Diagramme de déploiement du système	65
Figure 49: Model d'exploitation – définition des applications.....	66
Figure 50: Situation de la conception détaillée dans 2TUP.....	69
Figure 51: Micro-processus de conception détaillée	69
Figure 52: Définition des attributs des classes	70
Figure 53:Architecture en couches.....	71
Figure 54: L'arborescence de la couche présentation	72
Figure 55: L'arborescence de la couche Contrôleur.....	72
Figure 56: L'arborescence de la couche Métier	73
Figure 57: L'arborescence de la couche persistance	73
Figure 58: menu principale.....	80
Figure 59: la vue ajout rôle.....	80
Figure 60: la vue ajout rôle _ validation des champs de saisie.....	81
Figure 61: la vue ajout rôle _ validation des champs de saisie2.....	81
Figure 62: la vue consultation des rôles	82
Figure 63: la vue consultation des rôles _ modification d'un rôle	82
Figure 64: la vue consultation des rôles _ suppression d'un rôle	83
Figure 65: la vue d'ajout d'un poste.....	83
Figure 66: la vue consultation des postes	84
Figure 67: la vue d'ajout d'un intervenant	84
Figure 68: la vue de consultation des intervenants.....	85
Figure 69: la vue de consultation des intervenants _ création d'un nouveau compte	85
Figure 70: la vue de consultation des comptes	86
Figure 71: la vue de consultation des intervenants _ création d'un nouveau congé.....	86
Figure 72: la vue de consultation congés	87
Figure 73: la vue d'ajout d'une nouvelle activité.....	87
Figure 74: la vue d'ajout d'un nouveau type d'activité.....	88
Figure 75: la vue de consultation des activités	88
Figure 76: la vue d'ajout d'une nouvelle branche	89
Figure 77: la vue d'ajout d'une nouvelle branche	89
Figure 78: la vue d'ajout d'un nouveau projet	90
Figure 79: la vue de consultation des projets	90
Figure 81: la vue de consultation des projets après l'ajout d'une nouvelle branches à la liste des branches.....	91
Figure 82: la vue d'ajout d'un nouveau package.....	91
Figure 83: la vue de consultation des packages.....	92
Figure 84: la vue de consultation des packages_visualisation des informations d'un package	92
Figure 85: la vue de création d'une nouvelle livraison pour un package sélectionné	93
Figure 86: la vue de visualisation d'un package après sa livraison.....	93
Figure 87: la vue de création d'une demande des travaux pour un package sélectionné	94
Figure 88: la vue de consultation de la demande créée	94
Figure 89: la vue d'ajout d'un nouveau module.....	95
Figure 90: la vue de consultation de la liste des modules	95

Liste des abréviations



Abréviation	Désignation
UML	UnifiedModelingLanguage
2TUP	TwoTracksUnifiedProcess
MVC	Modèle-Vue-Contrôleur
JSF	JavaServer Faces
JPA	Java Persistence API
EJB	Enterprise JavaBeans
SGBD	Système de gestion de base de données
JEE	Java Enterprise Edition
Tus	Tests Unitaires
ZAM	Zodiac Aerospace Maroc
IHM	Interface Homme Machine
JDK	Java Development Kit

Liste des tableaux

Tableau 1 : Description du cas d'utilisation « Role ».....	35
Tableau 2 : Description du cas d'utilisation « Poste »	36
Tableau 3: Description du cas d'utilisation « Intervenant »	42
Tableau 4 : Description du cas d'utilisation « Activité »	38
Tableau 5 : Description du cas d'utilisation « Type d'activité ».....	39
Tableau 6: Description du cas d'utilisation « Branche »	45
Tableau 7: Description du cas d'utilisation « Projet ».....	41
Tableau 8: Description du cas d'utilisation « Package »	43
Tableau 9: Description du cas d'utilisation « Livraison »	43
Tableau 10:Description du cas d'utilisation « Congé ».....	44
Tableau 11: Description du cas d'utilisation « Compte »	45
Tableau 12: Concepts techniques du système	49
Tableau 13: classes du système par catégories	51
Tableau 14: Liste des IHM du système.....	68

Introduction générale

Actuellement, face à une concurrence de plus en plus rude, les entreprises cherchent désormais autant que possible à perfectionner leurs applications afin de les rendre de plus en plus rentables. Pour s'y faire elles essayent de les adapter aux nouvelles technologies et aux architectures logicielles les plus convenables.

Dans ce sens, les sociétés ont besoin d'une technologie qui rendra leurs systèmes légers, flexibles et faciles à maintenir, et d'une architecture qui leur permettra une bonne structuration de leurs systèmes en séparant les différentes logiques : présentation, métier et données.

Le design pattern MVC et l'architecture n-tiers viennent en fait pour répondre à ce besoin. Les atouts de ces technologies ont encouragé leur émergence et incité les entreprises à les adopter dans leurs développements. Ainsi, un nombre considérable de sociétés ont migré leurs applications vers ces technologies afin de tirer profit des multiples avantages qu'elles offrent.

A cet égard, et conformément à sa stratégie globale qui vise à améliorer le rendement de ses systèmes, Zodiac Aerospace Maroc a jugé nécessaire de mettre en place une application de gestion des projets en JEE6. C'est dans ce cadre que s'intègre mon projet de fin d'études qui vise le développement d'une application de gestion des projets, plus particulièrement des packages en utilisant les technologies web les plus avancées : MVC, JEE6 (JSF 2.0, primeFaces 3.2, EJB3, JPA 2).

Le présent rapport, comporte les trois parties suivantes :

Première partie « Contexte général du projet » : Composée de deux chapitres, le premier présente l'organisme d'accueil et le deuxième est une présentation générale du sujet.

Deuxième partie « Étude conceptuelle » : Présente les différentes phases du processus 2TUP. De l'étude préliminaire du projet jusqu'à la conception détaillée, sous forme de chapitres indépendants.

Troisième partie « Mise en œuvre du projet » : Décrit les différentes étapes de la mise en œuvre du projet à travers deux chapitres. Le premier expose les outils et technologies et le 2^{ème} présente les prototypages des interfaces.

Je termine ce rapport par une conclusion générale résumant les résultats et les perspectives de mon projet de fin d'études ainsi qu'une annexe dans laquelle j'ai fait la comparaison entre les différents processus de développement.

Partie I : Contexte général du projet



Cette première partie se veut être un prélude à l'étude menée dans ce projet.

Le premier chapitre intitulé « Présentation de l'organisme d'accueil » mets le point sur le groupe Zodiac Aerospace, son historique, son organigramme, ses activités et son capital humain.

Le deuxième chapitre fera l'objet d'une présentation générale du projet étudié dans ce présent rapport et finalement un aperçu sur la méthode et le planning suivie.

Chapitre I : Présentation de l'organisme d'accueil



L'objectif de ce chapitre est de fournir une présentation générale de la société Zodiac Aerospace en précisant, son historique, son organigramme, ses secteurs d'activités et son capital humain.



I.1. Le groupe Zodiac Aerospace

Présentation

La vocation de Zodiac pour l'aéronautique s'est faite jour en 1896 avec le développement des ballons dirigeables et des avions. Dans les années 1930, c'est l'invention du concept du bateau pneumatique qui permet à la société de prendre son essor et qui impose l'image de marque du Groupe après la seconde guerre mondiale. Aujourd'hui, le groupe est recentré sur l'aéronautique, après la cession de ses activités Marine en septembre 2007.

Zodiac Aerospace poursuit son développement pour assurer une croissance constante de ses activités. Avec une rigueur industrielle et de gestion, les savoir-faire technologiques de Zodiac sont organisés en trois métiers principaux : équipements de sécurité, systèmes embarqués, et intérieurs de cabines, et se déclinent en six branches d'activité. Elle compte 117 sites répartis dans le monde avec 17 540 employés contribuant au développement et à la croissance du groupe.

Organisation du groupe

En bonne logique avec son approche de développement sur des créneaux spécifiques et porteurs, les implantations industrielles du Groupe couvrent les zones géographiques où se concentrent les principaux constructeurs aéronautiques, notamment l'Amérique du Nord, l'Amérique du Sud et l'Europe.

Ainsi le Groupe peut-il offrir le meilleur service aux grands donneurs d'ordre tels que les avionneurs, les compagnies aériennes, les administrations, les armées, ...

Avec près des trois quarts de ses effectifs basés hors de France, Zodiac Aerospace bénéficie d'un esprit d'ouverture international indispensable à la compréhension des attentes de ses marchés et à la capacité de réagir et d'innover.

Métier du groupe

Le groupe Zodiac Aerospace exerce son métier dans six branches d'activités qui sont :

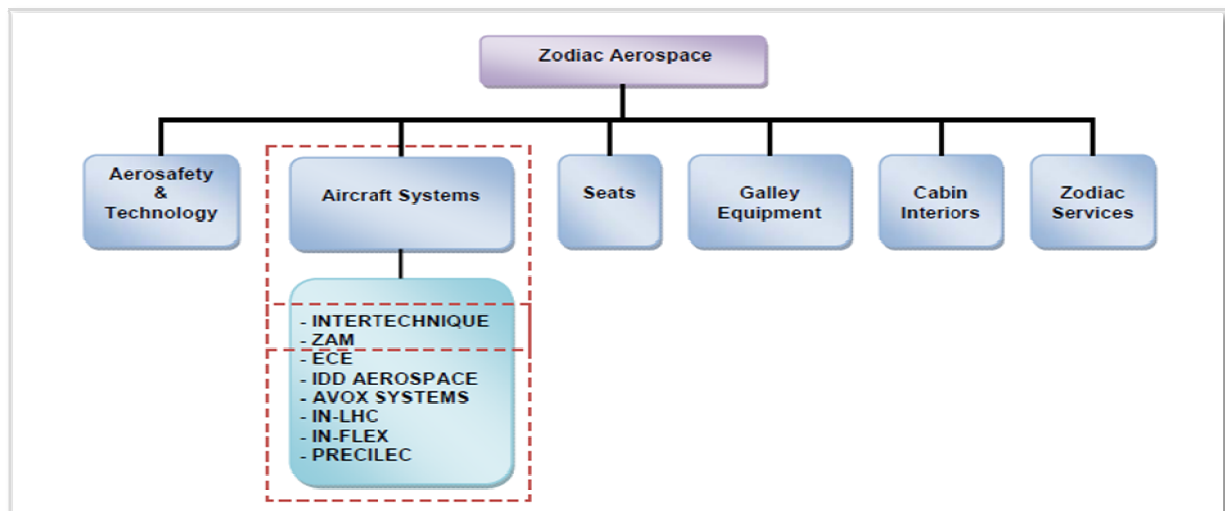


Figure 1 : Organisation en branches du groupe Zodiac Aerospace

Aerosafety&Technology

La branche Aerosafety&Technology conçoit et fournit des équipements, des produits, des systèmes complets dédiés au sauvetage et à la protection ainsi que des solutions technologiques pour des domaines d'applications très variés.

Sa mission a aussi pour vocation d'innover dans le domaine de la télémessure et de la télécommunication avec des produits et systèmes à forte valeur ajoutée pour les secteurs aéronautique et spatial tout comme pour certains marchés comme l'automobile avec ses solutions pour les airbags.

AircraftSystems

La branche AircraftSystems opère dans les parties citées ci-dessous:

- Circulation carburant,
- Surveillance et gestion de systèmes,
- Oxygène et protection physiologique,
- Gestion de la puissance électrique,
- Commandes et signalisation dans le cockpit,
- Eclairage et actionneurs,
- Hydraulique et régulation et
- Actionneurs, capteurs et moteurs.

Les sociétés de la Branche fournissent des équipements fiables et performants aux constructeurs d'avions et de moteurs et sont reconnues au plan mondial comme les spécialistes des équipements et des systèmes de haute technologie au service de fonctions essentielles tant en vol qu'au sol.

La branche développe des compétences de pointe dans les domaines suivants :

- Intertechnique : circulation carburant (circulation, gestion du carburant), surveillance et gestion de systèmes, calculateurs embarqués, surveillance et mesure de la température et de l'humidité.
- Intertechnique & AvonSystems : systèmes oxygène.
- ECE : gestion de la puissance électrique, commandes, signalisation dans le cockpit, éclairage et actionneurs.
- IN-LHC & IN-FLEX : hydraulique et régulation.
- PRECILEC : actionneurs, capteurs & moteurs.

Ce département a en charge l'étude et le développement des composants suivants :

- Capteurs : Capteur de jaugeage de carburant (jauge de réservoir), jauge d'huile moteur et de liquide hydraulique à capacitance (soit métallique, soit en matériau composite métallisé). Ces capteurs peuvent être équipés de têtes intégrant des circuits électroniques de traitement du signal.
- Capteurs de détection de niveau à capacitance ou à thermistance, fixés sur les parois des réservoirs ou sur les jauges de mesure. Ils sont utilisés pour le carburant et l'huile moteur.



Figure 2: Jauge de réservoir

- Capteur de mesure débitmétrique à hélice délivrant un signal proportionnel au débit volumique ou au débit massique.
- Capteur de mesure de température et d'humidité.
- Capteur de mesure d'épaisseur de givre.

Actionneurs :

- Actionneurs linéaires à moteurs électriques pas à pas et vis à billes pour vannes de prélèvement d'air.
- Actionneurs rotatifs de vannes à moteurs électriques continu et pas à pas.

Indicateurs :

Indicateurs à LCD et à aiguilles pour l'affichage de nombreux paramètres :

- Quantité de carburant et débit de carburant,
- Paramètres électriques,
- Température,
- Paramètres moteurs,
- Vitesse rotor et turbine libre,
- Epaisseur de givre,
- Paramètres de pressurisation;
- Vitesse de rotation.

Centrales d'indications combinant l'affichage à LCD de plusieurs paramètres, des voyants de signalisation et des interrupteurs.



Figure 3 : Indicateur à aiguilles

Calculateurs :

Ce sont essentiellement des calculateurs numériques temps réels qui assurent de nombreuses fonctions telles que :

- Mesure de paramètres,
- Régulation,
- Surveillance de paramètres de mesure et d'informations discrètes,
- Couplage à de nombreux types de Bus : ARINC 429, RS 232, 422 ou 485, MIL 1553, CAN, AFDX et
- Mémorisation d'informations pour l'aide à la maintenance.



Figure 4: Calculateur

Seats :

Cette activité est spécialisée dans la fabrication et la commercialisation de sièges pour les avions civils et les hélicoptères.

Cette activité permet à Zodiac Aerospace d'occuper la première place mondiale dans ce secteur, au travers de deux divisions, Sièges Europe et Sièges US, qui se caractérise chacune par une offre personnalisée et des gammes très spécifiques suivant les marchés.

Galley and Equipment :

L'activité Galley& Cargo Equipment assure la conception, la production et le service après-vente des galleys, des équipements de galley, en particulier des trolleys, et des équipements Cargo (containers) destinés aux avions commerciaux.

CabinInteriors :

La branche CabinInteriors intervient dans deux grands domaines d'activité, la conception d'équipements embarqués et l'aménagement d'intérieurs de cabine d'avions civils.

La branche se compose de deux divisions :

- La division Équipements de Cabine est principalement spécialisée dans le confort et l'aspect sanitaire de la vie à bord des avions. Elle réalise et fabrique principalement des équipements intégrés de Cabine tels que les systèmes de gestion de l'eau et des déchets, ou des éléments de galleys.
- La division Système cabine se positionne comme un leader dans l'aménagement de cabines d'avions en fournissant des solutions clé en mains aux avionneurs.

Zodiac Services :

Afin d'améliorer et d'optimiser son savoir-faire tout en restant à l'écoute de ses clients, le groupe a développé une activité service offrant un Service Clients ZODIAC SERVICES centralisé pour l'ensemble des produits du Groupe.

I.2. Zodiac Aerospace Maroc (ZAM)

Pourquoi choisir le Maroc ?

Les leaders mondiaux du secteur aéronaval envisagent une extension massive vers l'étranger. Ils ciblent les pays en cours de développement pour l'implantation de nouvelles unités industrielles. La concurrence est donc de plus en plus sévère et seuls les meilleurs

peuvent subsister. Dans un marché qui est totalement mondial, le Maroc profite de son contexte socio-économique favorable, pour devancer des concurrents régionaux très tenaces.

Plus de 30 ans d'expérience industrielle en aéronautique, proximité géographique et culturelle, disponibilité de compétences et mains d'œuvre hautement qualifiée sont tous des atouts qui font que le Maroc tient bon sur le terrain de compétition.

De nos jours l'enjeu est de taille ; le tissu industriel local continue à s'élargir. Plusieurs projets de développement soutiennent l'attractivité du pays. On cite entre autres :

- Le développement de la sous-traitance,
- L'ouverture sur les marchés internationaux
- L'engagement des établissements de formation dans la qualification des cadres et techniciens du secteur.

Tous ces facteurs font de la création d'une filiale de ZODIAC AEROSPACE au Maroc une vision à des perspectives stratégiques.

Présentation du ZAM

La société ZODIAC AEROSPACE Maroc, filiale du groupe ZODIAC AEROSPACE fait partie de la branche AIRCRAFT SYSTEMS. Elle s'organise en 5 entités comme le montre la figure 5, à savoir :

- Service Qualité et Sécurité,
- Administration Finances/Comptabilité,
- Production Mécanique,
- Production Electronique et
- Pôle ingénierie.

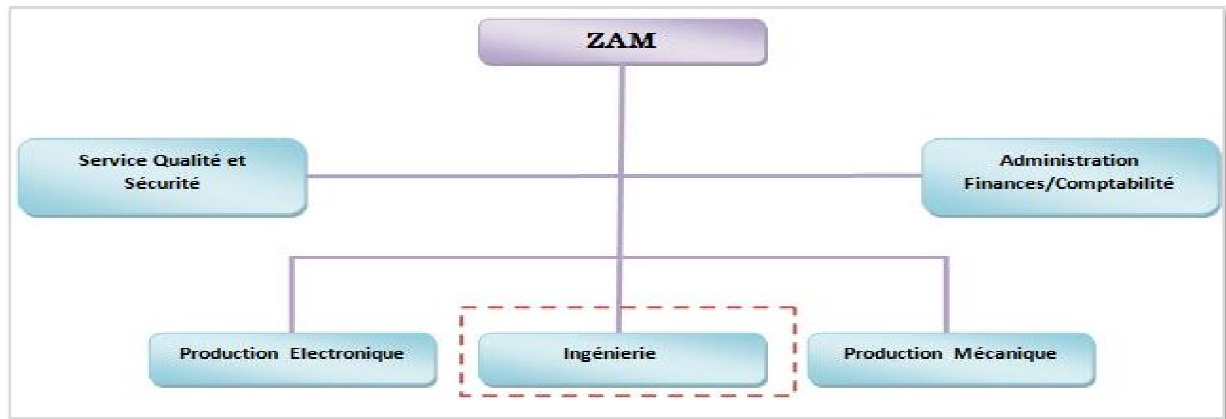


Figure 5: Organigramme de Zodiac Aerospace Maroc (ZAM)

Présentation du pôle ingénierie

Le département ingénierie est organisé en 2 pôles.

- Pôle Logiciel & FPGA : inclue toutes les équipes qui développent des logiciels.
- Pôle Bureau d'études : inclue toutes les équipes qui font le design des parties mécaniques ou électroniques.

Le pôle Logiciel & FPGA est constitué principalement de 4 équipes :

- D7 SW.
- ECE SW.
- ECE FPGA.
- Qualitéingénierie.

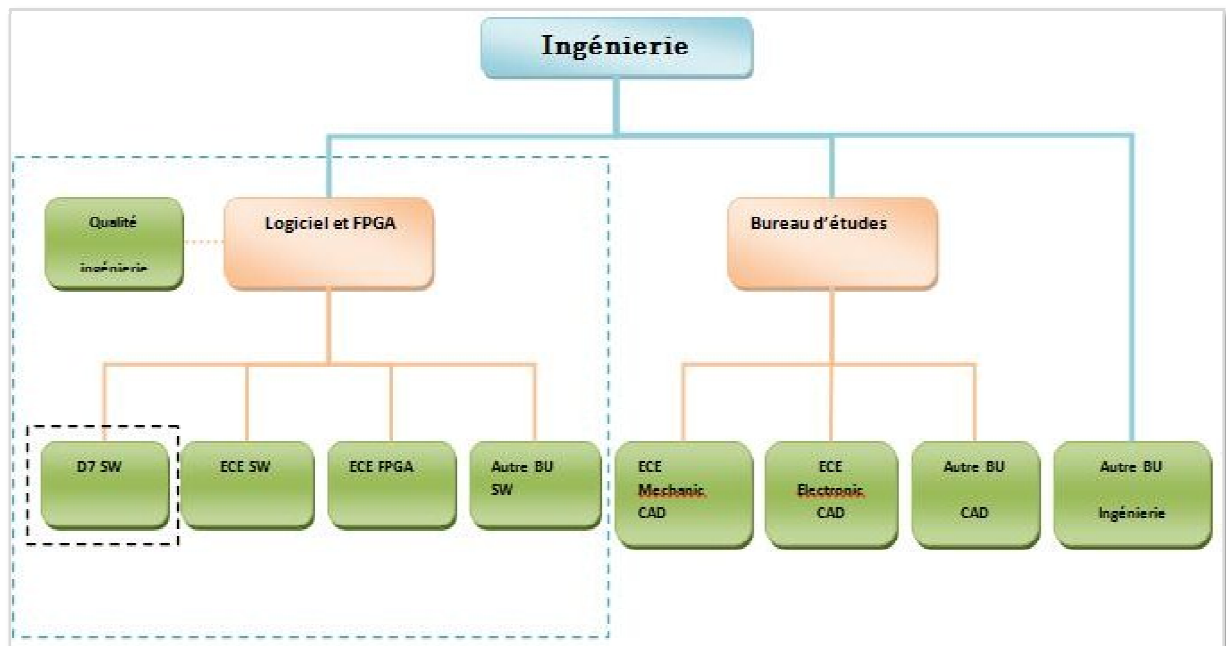


Figure 6 : Organigramme du pôle ingénierie du ZAM

I.3. Présentation de l'équipe de travail

Le stage s'est déroulé au sein de l'équipe D7 SW. Cette grande équipe, qui se compose de plus de 50 personnes, est divisé en quatre équipes :

- Tests Unitaires et Validation & vérification.
- Validation Scade et documents.
- Développement.
- Qualité et assurance des processus.

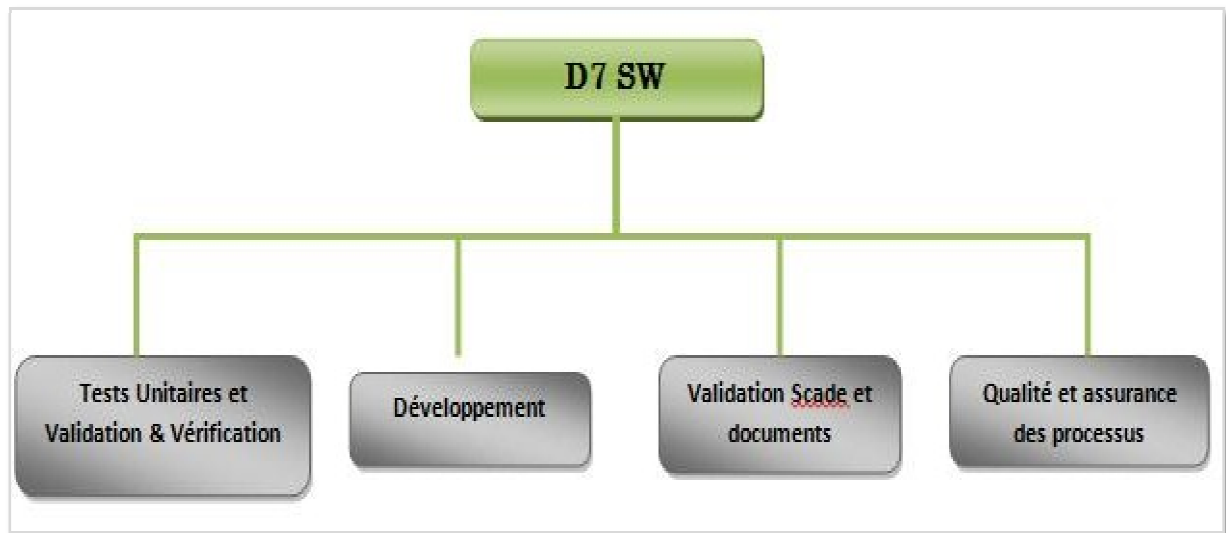


Figure 7 : Organigramme de l'équipe D7

I.4. Méthodologie et cycle de vie logiciel de l'équipe D7

Le cycle en V, peu adapté à la pratique lorsque la spécification du client évolue, est à la base de la méthodologie adoptée par INTERTECHNIQUE.

Cette méthodologie introduit une notion de développement incrémental avec des cycles différents selon l'objectif du logiciel :

- Logiciel prototype pour les tests sur banc ou au sol ;
- Logiciel prototype pour tests en vol ;
- Logiciel pour la certification DO-178B.

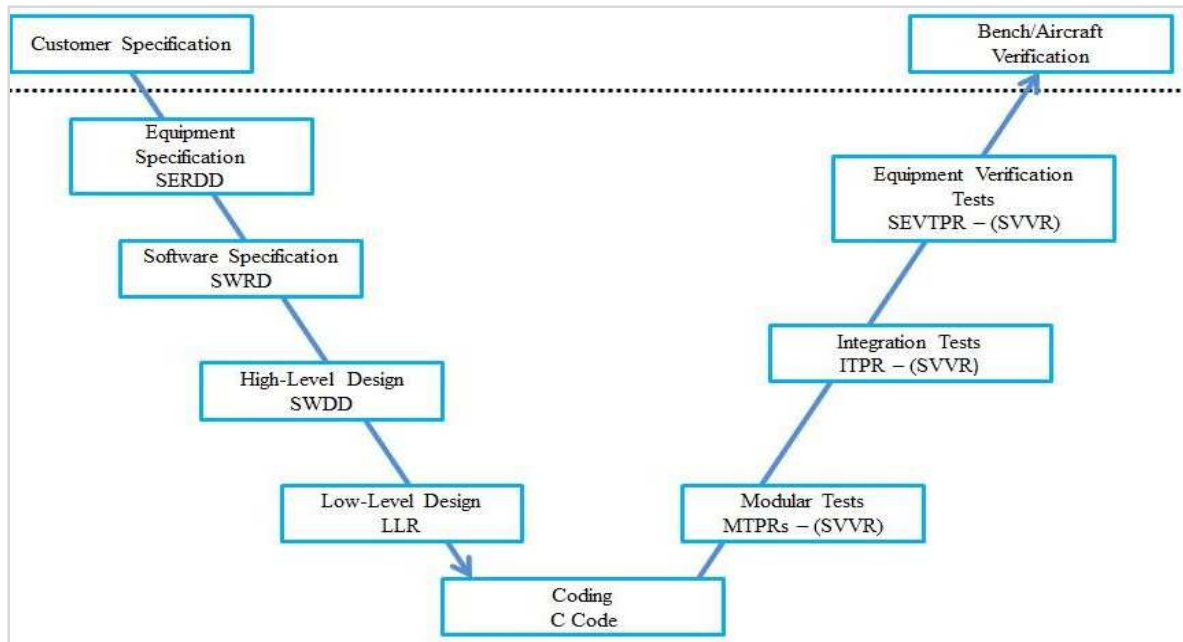


Figure 8 : Cycle en V - INTERTECHNIQUE.

Activités de développement

Planification :

- Définir et documenter le cycle de vie du logiciel et les processus applicables
- Estimer les efforts de développement ;
- Identifier les moyens à utiliser pour le développement (outils, équipements de test, formations . . .) ;
- Affecter les tâches et définir un planning.

Spécification :

- Développer les exigences fonctionnelles et opérationnelles de haut niveau à partir des exigences du système ;
- Développer les exigences de sécurité allouées au logiciel ;
- Décrire les critères de performance, les exigences du timing et les contraintes du design.

Conception de haut niveau :

- Développer les architectures statiques et dynamiques ;
- Identifier les modules traçables vers les exigences de haut niveau ou les exigences de sécurité ;
- Identifier les éventuels modules (exigences de bas niveau) dérivés ;
- Décrire la gestion des ressources, des interruptions . . .

Conception de bas niveau et codage :

- Pour tout module identifié dans le SWDD (Spécification), développer les exigences de bas niveau (LLR) à partir des exigences de haut niveau et des exigences de sécurité, ainsi que les LLR dérivés ;

- Produire le code source à partir de l'architecture du logiciel et des LLR.

Integration / debbuging :

- Produire des exécutables correspondant à une intégration graduelle des modules de l'architecture ;
- Charger et exécuter les exécutables dans la cible ; La phase d'intégration/debbuging est terminée lorsque le comportement du logiciel peut conduire à démarrer les tests formels.

Activités de Tests

❖ Tests modulaires :

- Vérifier que l'exécution des modules objets est conforme au LLR ;
- Vérifier les paramètres et le retour du module ;
- Vérifier le séquençement des appels des modules (control flow) ;
- Analyser et confirmer le degré de couverture structurelle.

❖ Tests d'intégration :

- Vérifier que le logiciel (code exécutable chargé dans l'équipement) interagit correctement avec le hardware ;
- Vérifier que les modules du logiciel interagissent correctement entre eux ;
- Vérifier l'architecture dynamique du logiciel et le control coupling ;
- Vérifier que le logiciel se comporte conformément aux HLR non testable pendant les tests de vérification système/équipement (boîte blanche /white box).

❖ Test de vérification système/équipement :

- Démontrer que le logiciel réalise complètement et correctement les fonctions demandées dans la spécification système/équipement. Les tests de vérification système/équipement sont réalisés dans l'équipement cible(boîte noire/black box).

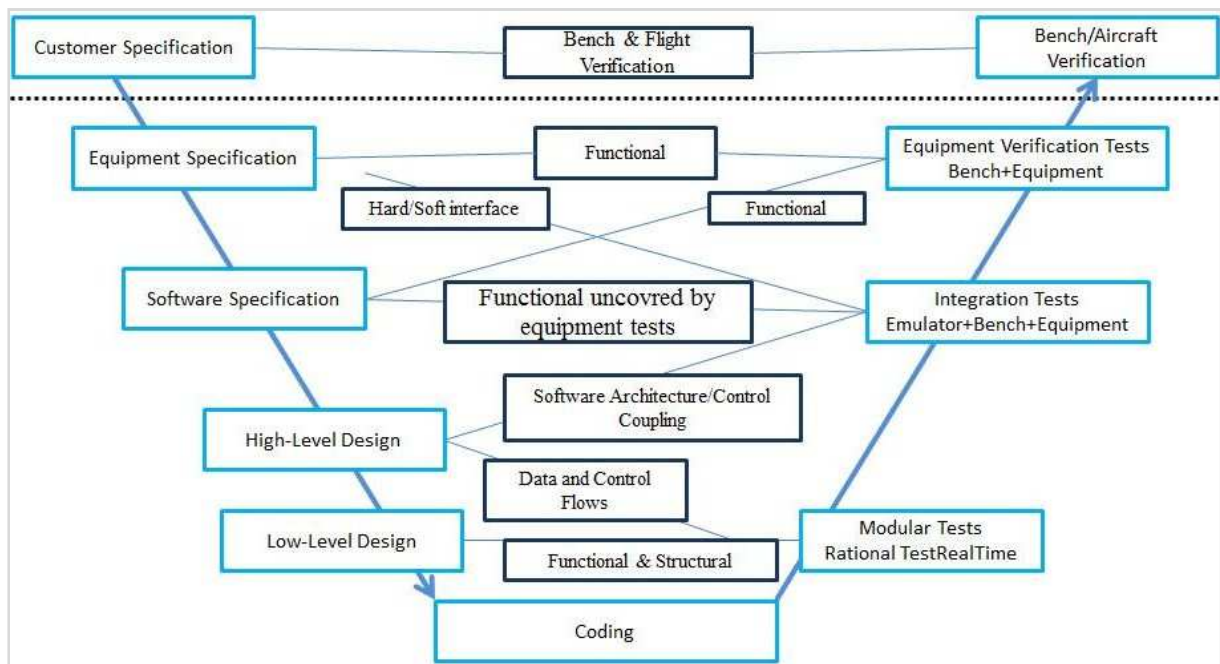


Figure 9 :Stratégie de Test - INTERTECHNIQUE.

Chapitre II : Présentation générale du sujet



L'objectif de ce chapitre est de donner un aperçu sur le projet réalisé, le planning de réalisation du projet et finalement le processus de développement et de modélisation.



II.1.Limites des méthodologies actuelles

Dans l'équipe Tests unitaires de Zodiac Aerospace Maroc (ZAM), la recherche de l'excellence est un souci constant. A cet effet les techniques de gestion, d'analyse et de synthèse des différentes activités utilisées jusqu'à présent permettent de dépenser beaucoup de temps pour avoir des résultats de la synthèse.

Toutefois, ces résultats restent insuffisants, la preuve est que malgré tous les efforts consentis, les synthèses sont difficilement conçues, et représentent des limitations.

Avant d'entamer le sujet, il faut savoir que le package est un ensemble des modules. Chaque module contient un code développé en C embarqué et constitue une partie d'un logiciel embarqué qui sera installé sur un avion. Pour ZAM, les projets sont des avions.

II.2. Présentation du Sujet

L'objectif principal est de mettre à disposition de ZAM une application de gestion, qui gère les packages, les livraisons, les congés et l'élaboration d'un tableau de bord destiné à exploiter les différentes informations générées par l'équipe de TU/VALIDATION de l'entreprise. Afin de répondre aux besoins des clients aussi bien au niveau stratégique qu'opérationnel.

L'application principale sera utilisée par l'ensemble des intervenants sur le projet, elle doit donc être fiable, conviviale et facile à manipuler. En effet, son utilisation ne devra pas laisser place à d'éventuels points faibles, et devra reprendre toute la démarche de la gestion effectuée.

II.3. Composantes de la solution

Pour mettre en place une solution de tableaux de bord on distingue trois phases différentes que la solution doit gérer :

La collecte des données

Le système d'information est la clé de voûte pour disposer d'une information correcte et fiable. C'est pourquoi il est important d'intégrer la solution dans le système d'information et de la faire communiquer avec les applications métiers existantes.

L'analyse et la restitution des données

L'outil recueille la requête et produit les indicateurs voulus. Ces fonctionnalités diffusent et présentent les informations de la façon la plus opérationnelle possible pour les différentes populations d'utilisateurs. La restitution graphique doit être ergonomique et personnalisable.

Les indicateurs de performances

Véritables témoins des processus de l'entreprise et de leurs tendances, les indicateurs de performances sont indispensables à la compréhension des situations et à l'anticipation dans les prises de décisions stratégiques et opérationnelles.

II.4. Conduite de projet

Planification du projet

Conduire un projet, c'est assurer le pilotage d'un processus de changement avec des ressources Dédiées en optimisant les compétences, l'organisation, les systèmes et les outils de conduite.

Une approche managériale réactive, souple et systématique pour mener à bien des changements Importants, complexes, ciblés sur le but à atteindre. Il y a trois niveaux de gestion du projet : La gestion de la production, des ressources et du temps.

Afin de satisfaire à ce dernier critère qu'est la gestion du temps, il est nécessaire d'établir un planning prévisionnel. Sachant que la ressource affectée à ce projet est moi et que le présent planning est réajusté au fur et à mesure de l'avancement du projet.

Le stage débute le 13 février et prend fin le 13 juin, ce qui implique une durée totale de 85 J*H selon le planning suivant :

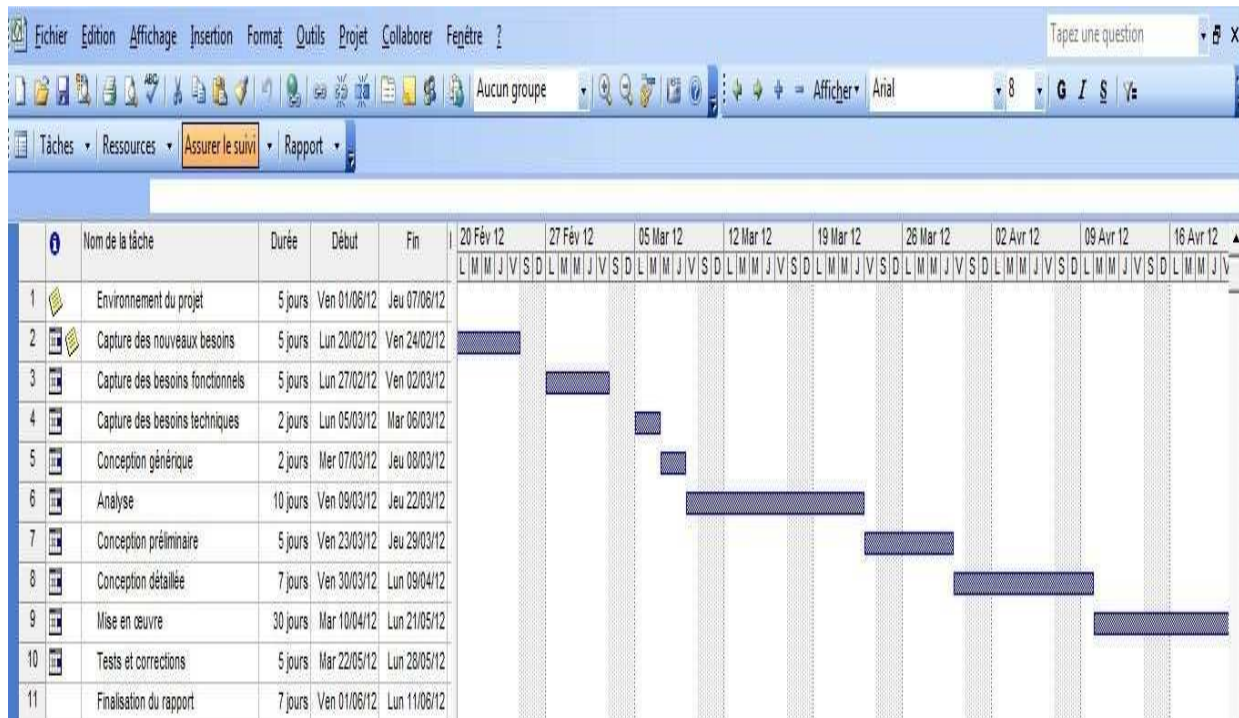


Figure 10: Planning du projet

Processus de développement

Choix du processus :

Le processus de développement constitue un facteur déterminant dans la réussite d'un projet, du fait qu'il cadre ses différentes phases et caractérise les principaux traits de sa conduite. Pour cela, le choix d'une méthode de développement adéquate aux exigences fonctionnelles, techniques et qualitatives d'un projet, doit être élaboré au préalable afin d'obtenir un produit qui répond aux besoins et attentes des utilisateurs. Etant donné que chaque type d'application en informatique (développement spécifique, migration, intégration, maintenance ...), présente des exigences particulières en termes de méthodologie et de démarche à suivre. C'est pour cela, j'ai commencé par une étude comparative entre les processus les plus populaires. Mon objectif est de sélectionner le processus le plus adapté au contexte et aux objectifs de mon projet (Annexe A).

En se basant sur les résultats de cette étude comparative des processus, on peut constater le caractère itératif de toutes les méthodologies, que ce soit au niveau des plannings, des spécifications, ou du développement.

Le processus 2TUP reste un choix adapté à mon projet permettant de prendre en compte la séparation et l'amélioration du côté technologique et fonctionnel en réservant des branches entières à la technologie et au fonctionnel dans son processus de développement.

Processus de développement 2TUP

Le processus 2TUP propose un cycle de développement en Y en dissociant les aspects techniques des aspects fonctionnels comme l'illustre la figure suivante :

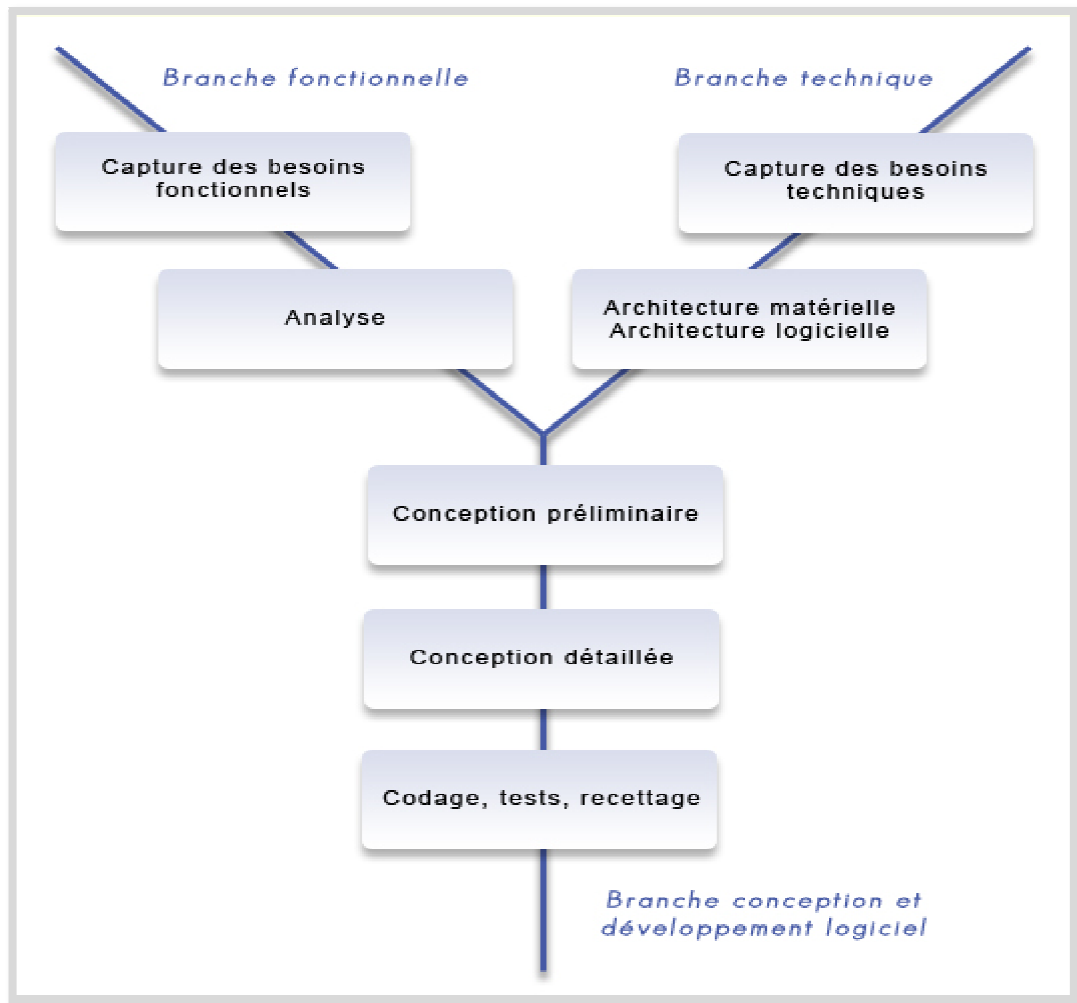


Figure 11 : Le processus de développement 2TUP

Branche fonctionnelle :

Les principales étapes de la branche fonctionnelle se présentent comme suit :

L'étape capture des besoins fonctionnels produit le modèle des besoins focalisé 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 boîte noire, c'est le niveau cas d'utilisation.

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 terme de métier.

Branche technique :

Les principales étapes de la branche technique se présentent comme suit :

L'étape 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.

L'étape 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. Ces derniers, délivrant les services techniques, assurent la réponse aux exigences opérationnelles du système.

Branche conception – réalisation :

Les principales étapes de cette branche se présentent comme suit :

L'étape conception préliminaire : étape qui 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.

L'étape 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.



Processus de modélisation UML

Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de part leur facilité et clarté, de bien modéliser le système à chaque étape.

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage, c'est pour ça qu'UML est présenté parfois comme une méthode alors qu'il ne l'est absolument pas.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage.

Voici une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet :

Le diagramme des cas d'utilisation : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.

Le diagramme de packages : présent depuis UML 2.0, ce diagramme modélise des catégories cohérentes entre elles, pour un souci de partage des rôles. Correspond à l'étape de modélisation des différents scénarios d'un cas d'utilisation.

Le diagramme de classes : sûrement l'un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet.

Le diagramme de séquence : représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système.

Partie II : Etude conceptuelle



Cette partie représente le noyau de ce présent rapport, elle regroupe la partie étude des besoins, l'analyse et la conception.

Le premier chapitre intitulé Etude préliminaire a pour objectif de mettre en évidence les besoins du client en rédigeant un cahier des charges et en identifiant les acteurs et les messages du système.

La capture des besoins fonctionnels consiste à recenser et détailler les différents cas d'utilisation du système ainsi d'illustrer les différents enchainements.

La capture des besoins techniques mettra le point sur la spécification logicielle et matérielle du futur système.

Le quatrième chapitre est celui de l'analyse son objectif est de sortir avec un digramme de classes complet et valide.

Les derniers chapitres représentent la partie de la conception qui a pour objectif de définir les frameworks techniques et leur organisation dans le système, et le prototype technique.

Chapitre I : Etude préliminaire et capture des besoins fonctionnels



Ce chapitre va me servir à poser les bases de la capture des besoins du système à réaliser.

Je vais commencer par l'étude préliminaire où je vais détailler le cahier des charges, puis je vais passer à l'étape de capture des besoins fonctionnels où je vais identifier et détailler les diagrammes des cas d'utilisations .



I. Etude préliminaire

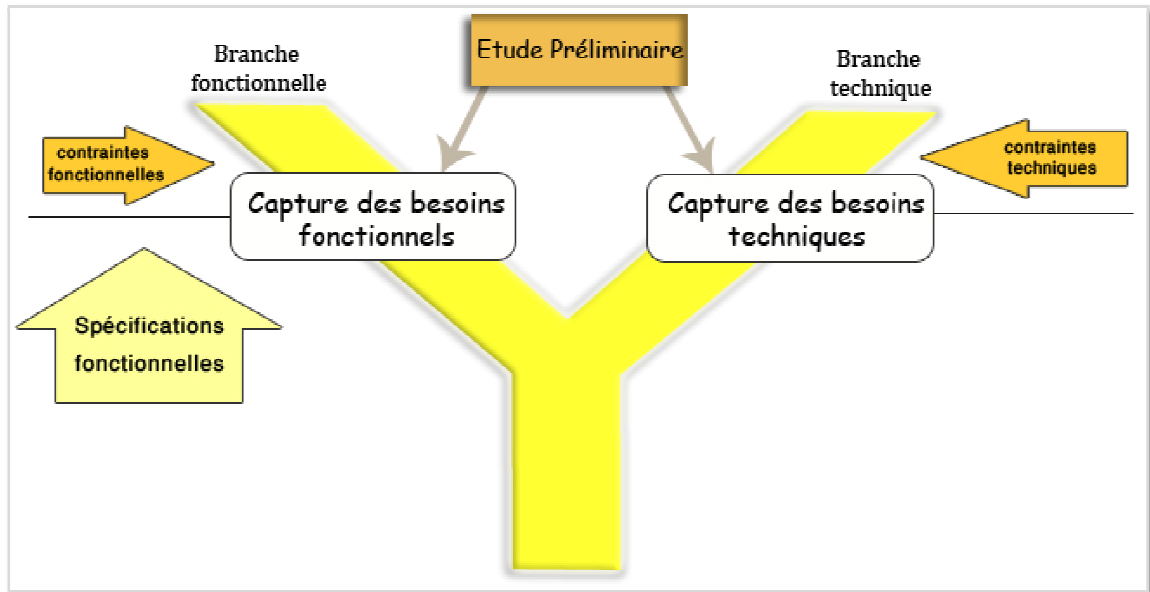


Figure 12: Situation de l'étude préliminaire dans 2TUP

I.1. Besoins fonctionnels

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 pour lui. Seul le roll-out (Administrateur) qui aura le droit d'accéder à toutes les fonctionnalités de l'application (visionner, ajouter, modifier ou supprimer). Les utilisateurs hors l'équipe TUs auront seulement le droit de visualiser les fonctionnalités et les informations de l'application.

Gestion des packages

Une phase d'authentification est nécessaire pour pouvoir accéder à la zone de saisie des packages. Un package peut être ajouté seulement de la part d'un roll-out, qui peut être aussi consulté, modifié ou supprimé. Une conservation de la même hiérarchie que celle de l'espace de travail (Branche->Projet->Package). La suppression ou la modification d'un package ou d'un composant du package (ressources, complexité ...), doit être validé par le roll-out.

Gestion des livraisons

Une phase d'authentification est nécessaire pour pouvoir accéder à la zone de consultation des livraisons. Pour la création d'une livraison, sa modification ou sa suppression il faut avoir le rôle administrateur afin d'avoir ces droits.

Recherche par Période

Cette fonctionnalité permet de rechercher tous les travaux effectués pendant une période sélectionnée par l'utilisateur.

Recherche par Package

Cette fonctionnalité permet de rechercher le package à partir de son nom à partir d'un champ de saisie.

Recherche par Intervenant

La fonctionnalité permet de rechercher le ou les packages par nom d'intervenant.

Recherche par Projet

Cette fonctionnalité permet à l'utilisateur de savoir tous les packages qui sont en cours ou sont déjà terminé sur un projet sélectionné

Calcule des indicateurs

Cette fonctionnalité permis de calculer les différentes métriques de l'équipe :Nombre de procédure développé par jour par homme...

I.1.1. Identification des acteurs

Les acteurs qui interviennent dans notre système sont humains, ils représentent toute personne ayant accès à l'application sachant que l'application est en architecture client /serveur ,ça sera installer sur une machine et accessible via les autres machines connectées sur le réseaux .

Selon les droits d'accès qu'on a défini ,on trouve deux acteurs principaux ,qui sont :

L'administrateur : qui est le chef d'équipe mais ça peut être le référent technique ou n'importe quel intervenant que le chef d'équipe veut lui accorder tous les droits.

L'utilisateur : les utilisateurs qui ont un accès restreint à l'application, généralement ils ont le droit juste de consultation, les opérations telles que l'ajout, la modification, la suppression ne sont pas autorisées pour ce genre d'acteurs.

I.1.2 Identification des messages

Le système émet :

- Liste des intervenants qui ont accès à l'application
- Liste des congés existants.
- Liste des packages.
- Liste des projets
- Liste des branches
- Liste des livraisons.
- Liste des comptes utilisateurs.
- Liste des rôles existants.
- Liste des postes existants.
- Liste des activités.
- Tableaux de bords

Le système reçoit :

- Informations sur les différentes entités citées dans ce que le système émette.

II.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 adéquat pour la modélisation de 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.

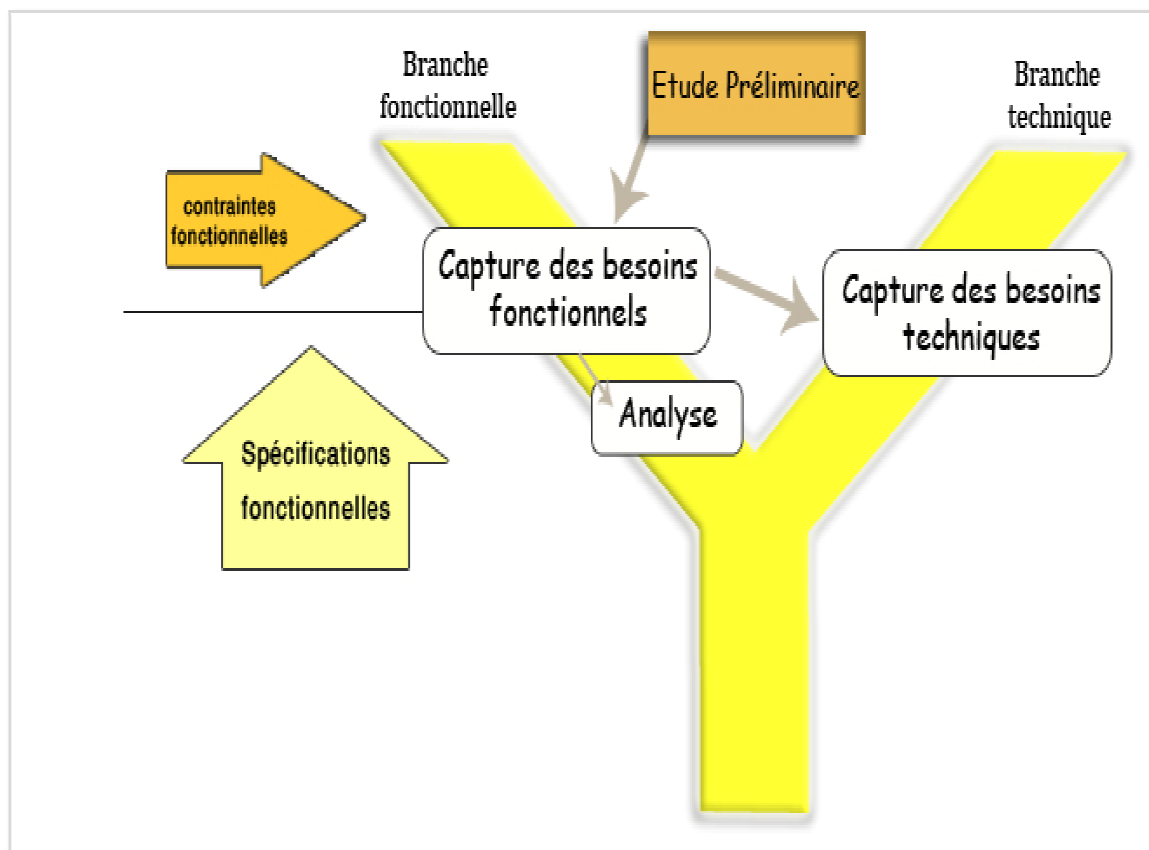


Figure 13: Situation dans 2TUP

II.1. Identification et description des Cas d'utilisations

Diagramme des Cas d'utilisation pour l'entité « Rôle » :

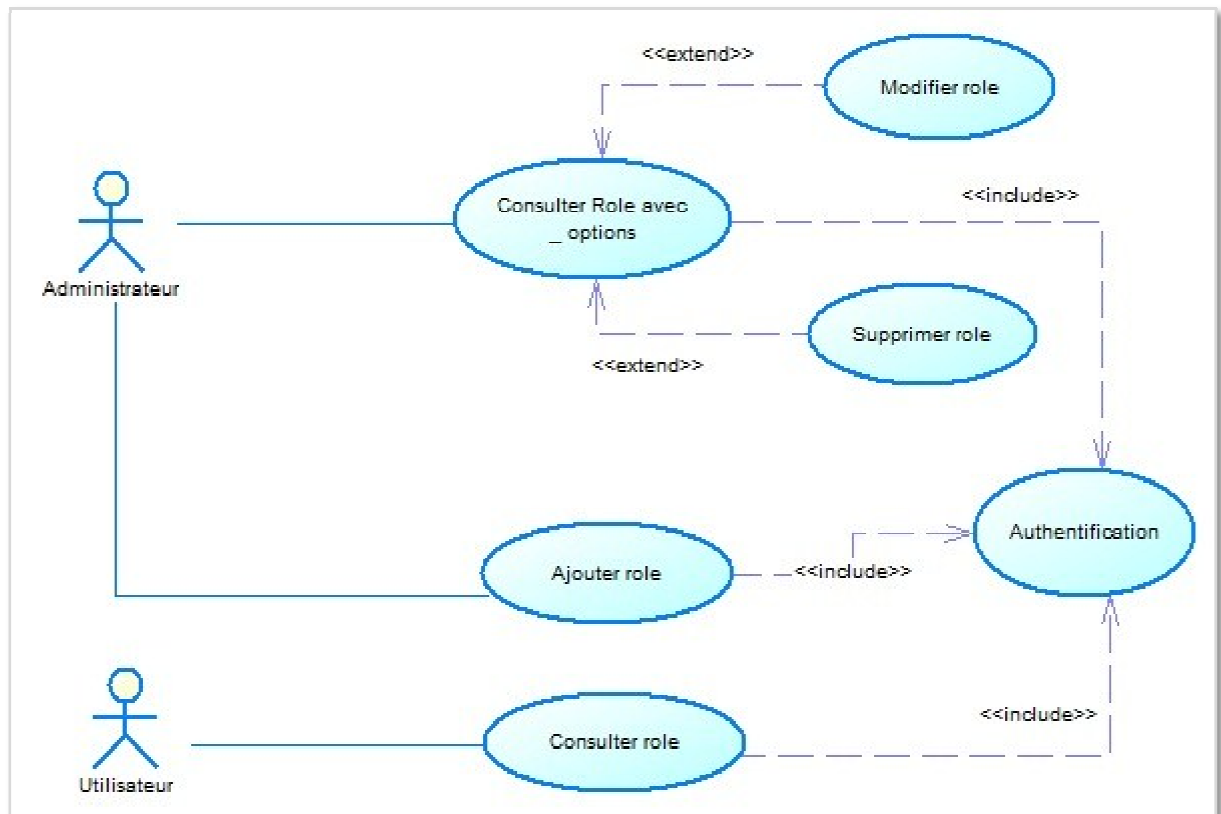


Figure 14: Diagramme de cas d'utilisation « Rôle »

Description du diagramme du cas d'utilisation pour l'entité « Rôle » :

Libellé	Acteur	Description
Ajouter Rôle	Administrateur	Création d'un nouveau rôle qui va nous aider dans la partie de la gestion des droits d'accès.
Modifier Rôle	Administrateur	Modification des champs du rôle sélectionné
Supprimer Rôle	Administrateur	Suppression du rôle sélectionné
Consulter Rôle	Administrateur	Affichage des différents rôles qui sont insérés dans la base de données dans un tableaux

Tableau 1 : Description du cas d'utilisation « Role »

Diagramme des Cas d'utilisation pour l'entité « Poste » :

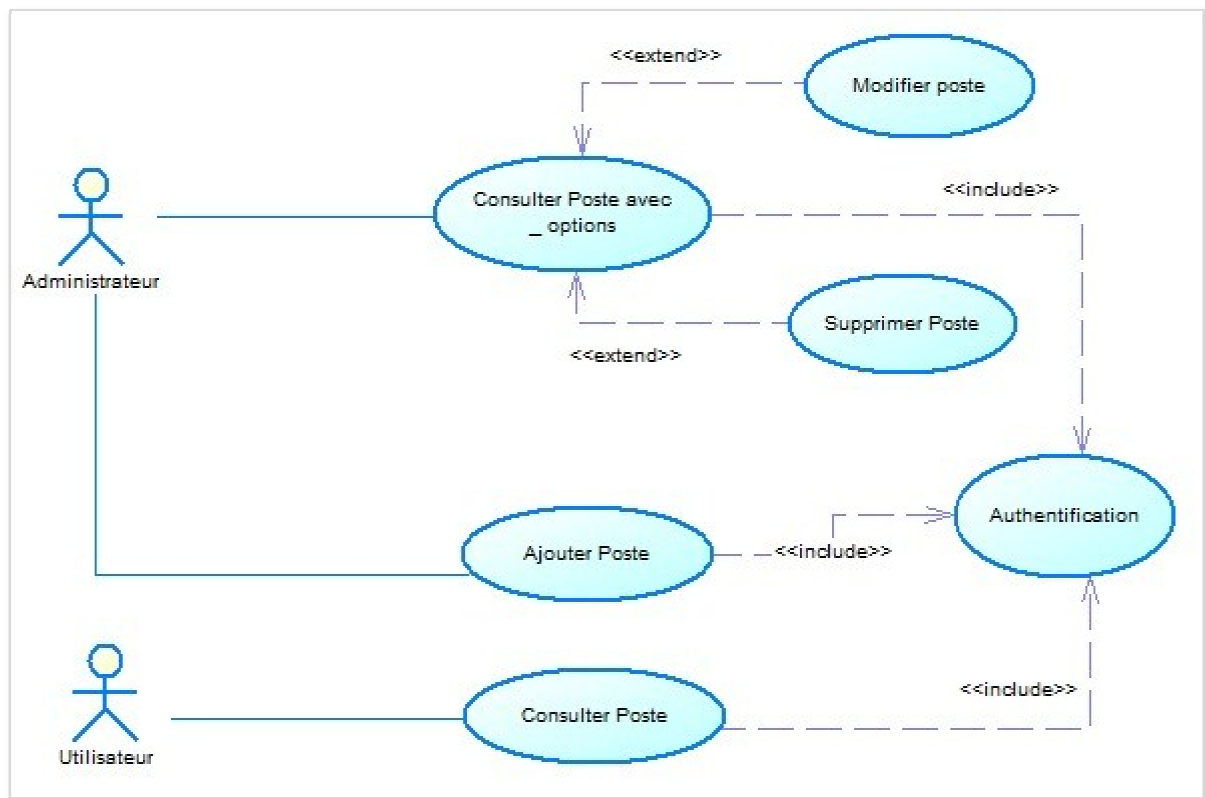


Figure 15: Diagramme de cas d'utilisation « Poste »

Description du diagramme du cas d'utilisation pour l'entité « Poste » :

Libellé	Acteur	Description
Ajouter Poste	Administrateur	Création d'un nouveau poste qui est bien une des fonctions des membres d'équipe.
Modifier Poste	Administrateur	Modification du poste sélectionné
Supprimer Poste	Administrateur	Suppression du poste sélectionné
Consulter Poste	Administrateur Utilisateur	Affichage des différents postes qui sont insérés dans la base de données dans un tableaux

Tableau 2 : Description du cas d'utilisation « Poste »

Diagramme des Cas d'utilisation pour l'entité « Intervenant » :

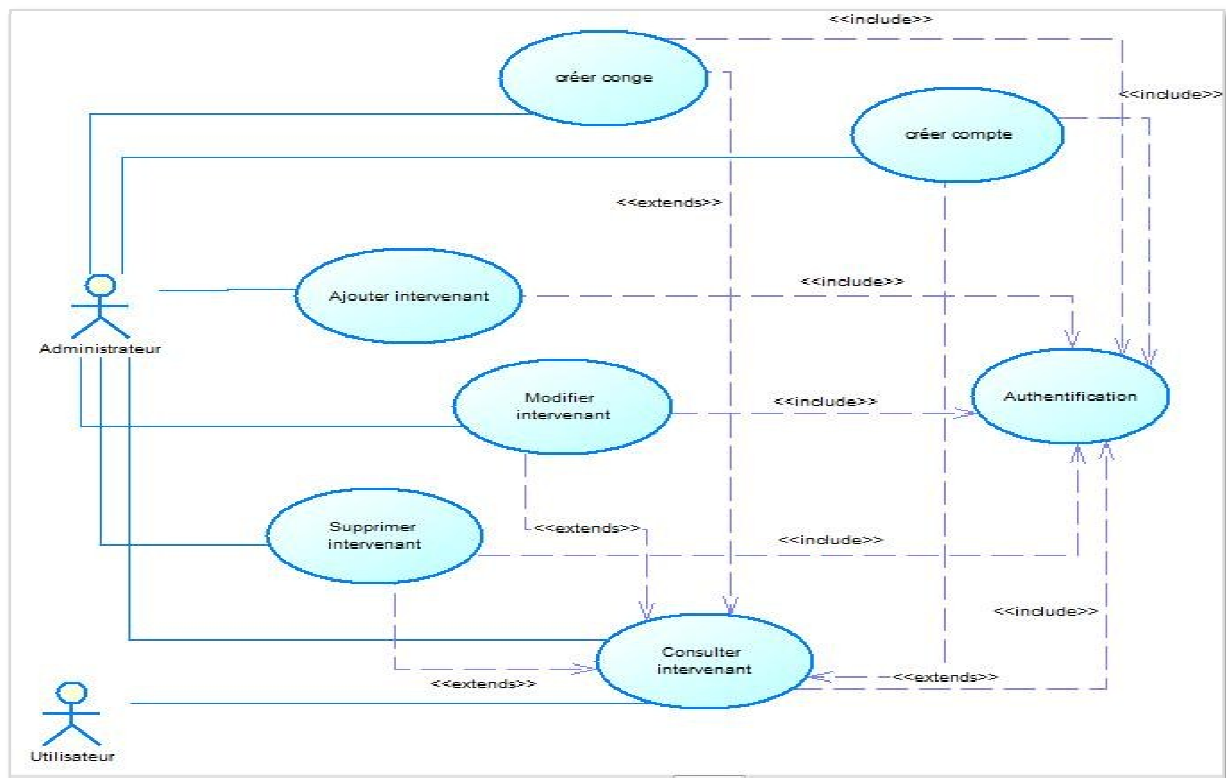


Figure 16: Diagramme de cas d'utilisation « Intervenant »

Libellé	Acteur	Description
Ajouter Intervenant	Administrateur	Création d'un nouveau intervenant en remplissant un formulaire qui contient toutes les informations nécessaires de l'intervenant.
Modifier Intervenant	Administrateur	Modification de l'intervenant sélectionné
Supprimer Intervenant	Administrateur	Suppression de l'intervenant sélectionné
Consulter Intervenant	Administrateur Utilisateur	Affichage des différents intervenants qui sont insérés dans la base de données dans un tableaux
Créer compte	Administrateur	L'administrateur peut créer des comptes pour les intervenants existants directement de la page de consultation pour que cet intervenant puisse s'authentifier et accéder à l'application.
Créer congé	Administrateur	L'administrateur peut créer un nouveau congé et l'affecter à l'intervenant convenable à partir de la page de consultation

Tableau 3: Description du cas d'utilisation « Intervenant »

Diagramme des Cas d'utilisation pour l'entité « Activité » :

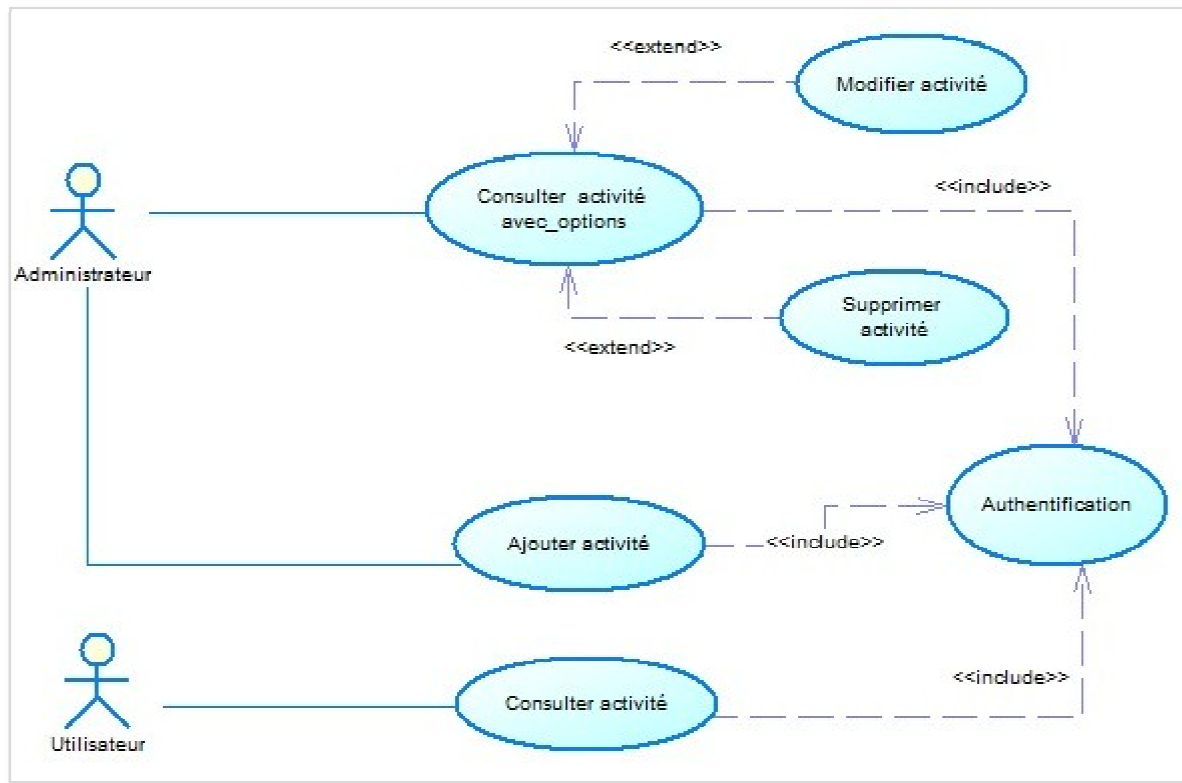


Figure 17: Diagramme de cas d'utilisation « Activité »

Description du diagramme des cas d'utilisations pour l'entité « Activité » :

Libellé	Acteur	Description
Ajouter Activité	Administrateur	Création d'une nouvelle activité .
Modifier Activité	Administrateur	Modification de l'activité concernée.
Supprimer Activité	Administrateur	Suppression de l'activité concernée.
Consulter Activité	Administrateur Utilisateur	Affichage des différentes activités qui sont insérées dans la base de données dans un tableaux

Tableau 4 : Description du cas d'utilisation « Activité »

Diagramme des Cas d'utilisation pour l'entité « Type d'activité » :

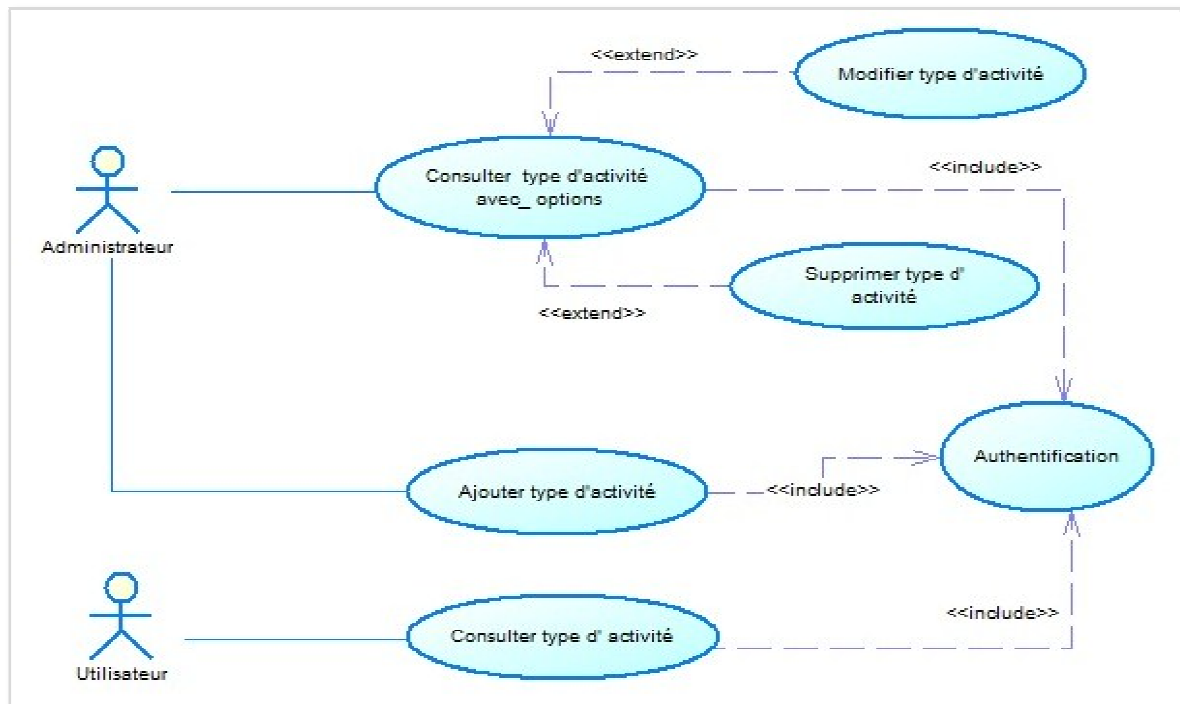


Figure 18: Diagramme de cas d'utilisation « Type d'activité »

Description du diagramme de cas d'utilisation pour l'entité :« Type d'activité »

Libellé	Acteur	Description
Ajouter Type d'activité	Administrateur	Création d'un nouveau type d'activité .
Modifier Type d'activité	Administrateur	Modification du type d'activité concerné.
Supprimer Type d'activité	Administrateur	Suppression du type d'activité concerné.
Consulter Type d'activité	Administrateur Utilisateur	Affichage des différentes types d'activités qui sont insérées dans la base de données dans un tableau

Tableau 5 : Description du cas d'utilisation « Type d'activité »

Diagramme des Cas d'utilisation pour l'entité « Branche » :

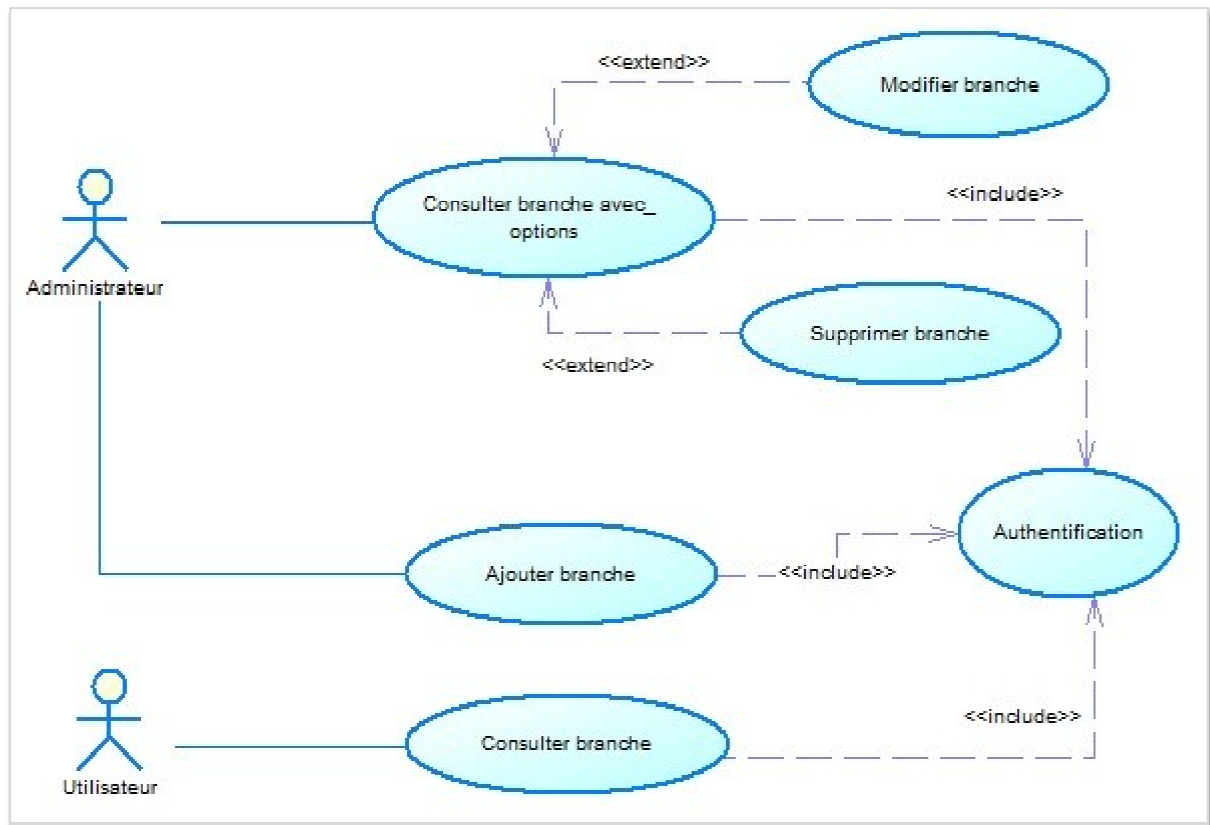


Figure 19: Diagramme de cas d'utilisation « Branche »

Description du diagramme des cas d'utilisation pour l'entité « Branche » :

Libellé	Acteur	Description
Modifier branche	Administrateur	Modification de la branche concernée.
Supprimer branche	Administrateur	Suppression de la branche concernée.
Consulter branche	Administrateur Utilisateur	Affichage des différentes branches qui sont insérées dans la base de données dans un tableaux

Tableau 6 : Description du cas d'utilisation « Branche »

Diagramme des Cas d'utilisation pour l'entité « Package » :

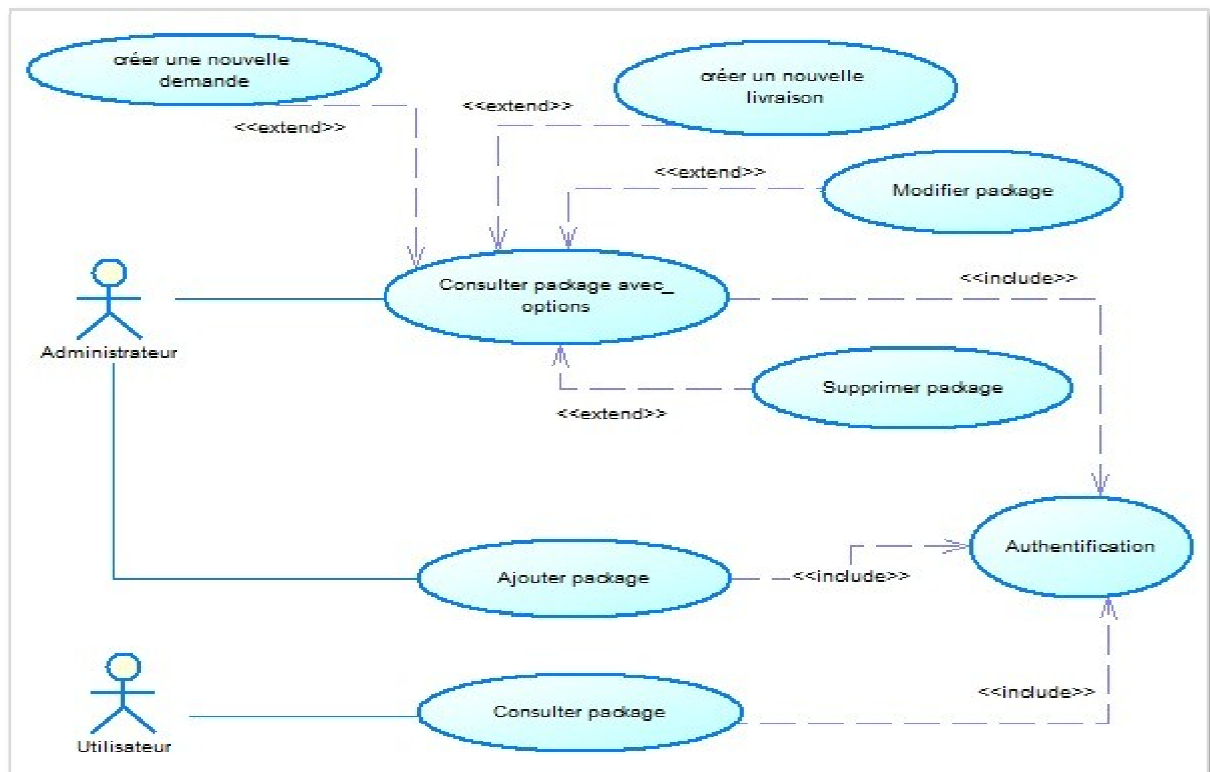


Figure 21: Diagramme de cas d'utilisation « Package »

Description du diagramme du cas d'utilisation pour l'entité « Package» :

Libellé	Acteur	Description
Ajouter package	Administrateur	Création d'un nouveau package.
Modifier package	Administrateur	Modification du package concerné.
Supprimer package	Administrateur	Suppression du package concerné.
Consulter package	Administrateur Utilisateur	Affichage des différents packages qui sont insérés dans la base de données dans un tableaux
Créer une demande des travaux	Administrateur	Création de la demande des travaux qui sera par la suite envoyée aux différents intervenants qui vont travailler sur la

		package concerné.
Créer une livraison	Administrateur	Création de la livraison qui se fait à la fin de la clôture de chaque package.
Rechercher package	Administrateur Utilisateur	Rechercher les packages par nom du package, par projet, par intervenant et par période.

Tableau 8: Description du cas d'utilisation « Package »

Diagramme des Cas d'utilisation pour l'entité « Livraison » :

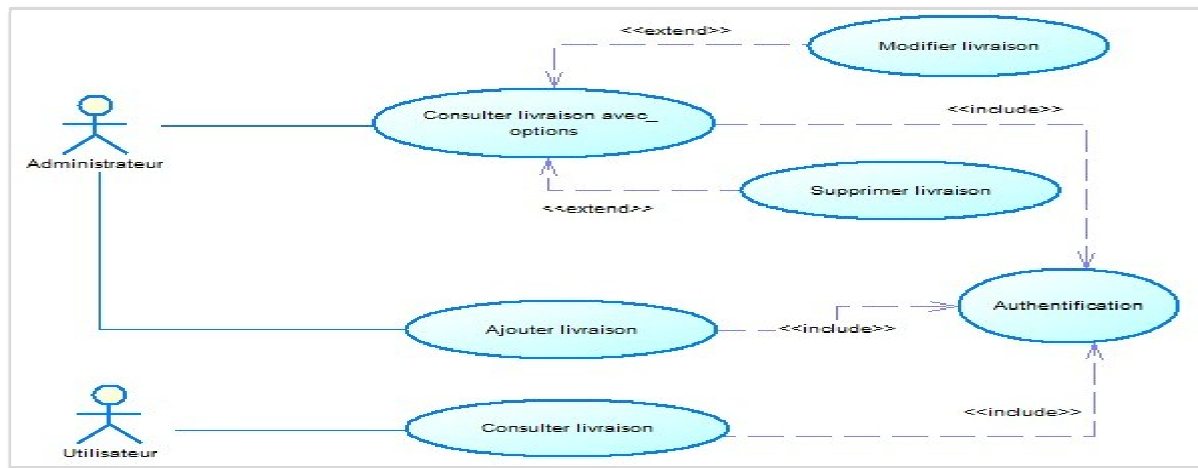


Figure 22: Diagramme de cas d'utilisation « Livraison »

Description du diagramme du cas d'utilisation pour l'entité « Livraison » :

Libellé	Acteur	Description
Modifier Livraison	Administrateur	Modification de la livraison concernée.
Supprimer livraison	Administrateur	Suppression de la livraison concernée.
Consulter livraison	Administrateur	Consultation des différentes livraisons existantes dans la base de données.

Tableau 9: Description du cas d'utilisation « Livraison »

Diagramme des Cas d'utilisation pour l'entité « Congé » :

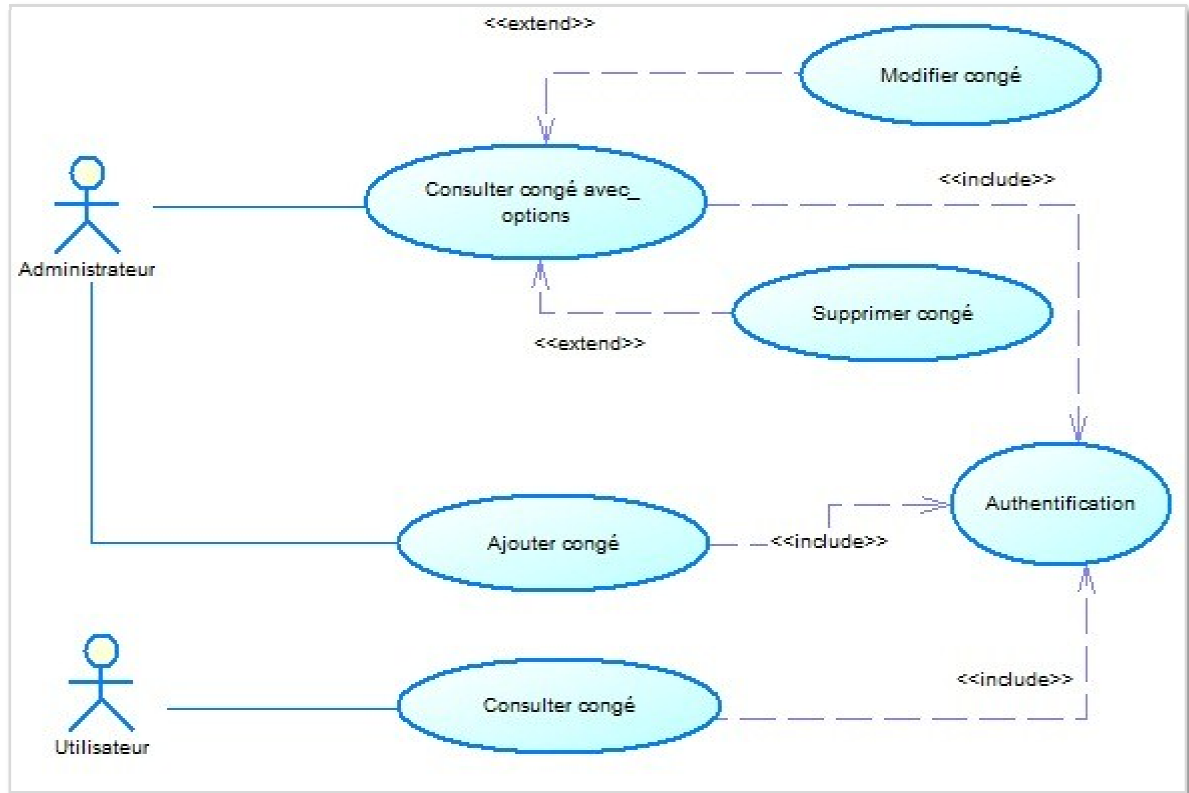


Figure 23: Diagramme de cas d'utilisation « Congé »

Description du diagramme du cas d'utilisation pour l'entité « Congé » :

Libellé	Acteur	Description
Modifier Congé	Administrateur	Modification du congé concerné.
Supprimer Congé	Administrateur	Suppression du congé concerné.
Consulter Congé	Administrateur Utilisateur	Consultation des différents congés existants dans la base de données.

Tableau 10:Description du cas d'utilisation « Congé »

Diagramme des Cas d'utilisation pour l'entité « Compte » :

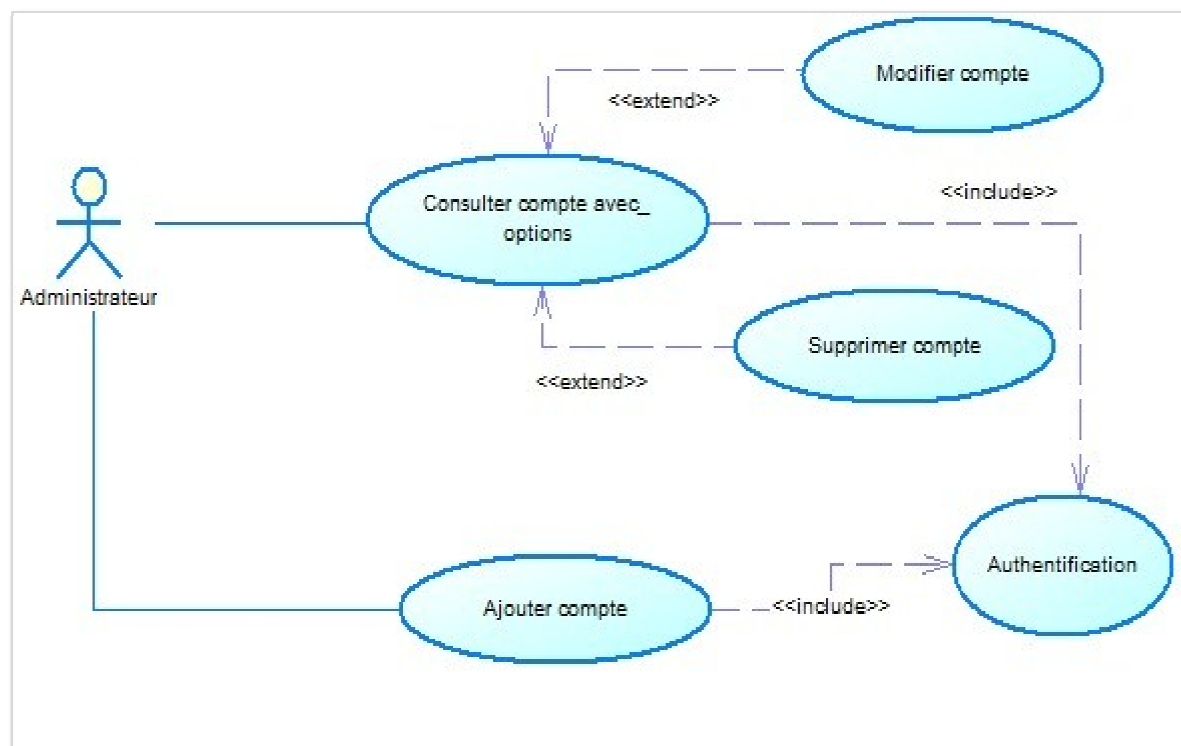


Figure 24: Diagramme de cas d'utilisation « Compte »

Description du diagramme du cas d'utilisation pour l'entité « Compte » :

Libellé	Acteur	Description
Modifier compte	Administrateur	Modification du compte concerné.
Supprimer compte	Administrateur	Suppression du compte concerné.
Consulter compte	Administrateur Utilisateur	Consultation des différents comptes existants dans la base de données.

Tableau 11: Description du cas d'utilisation « Compte »

II. 2. Identification des classes candidates

Le diagramme des classes candidates est un premier diagramme de classes obtenu à partir de l'identification des cas d'utilisation. C'est un premier travail d'analyse qui permet une première abstraction du système sous forme d'objets et de classes.

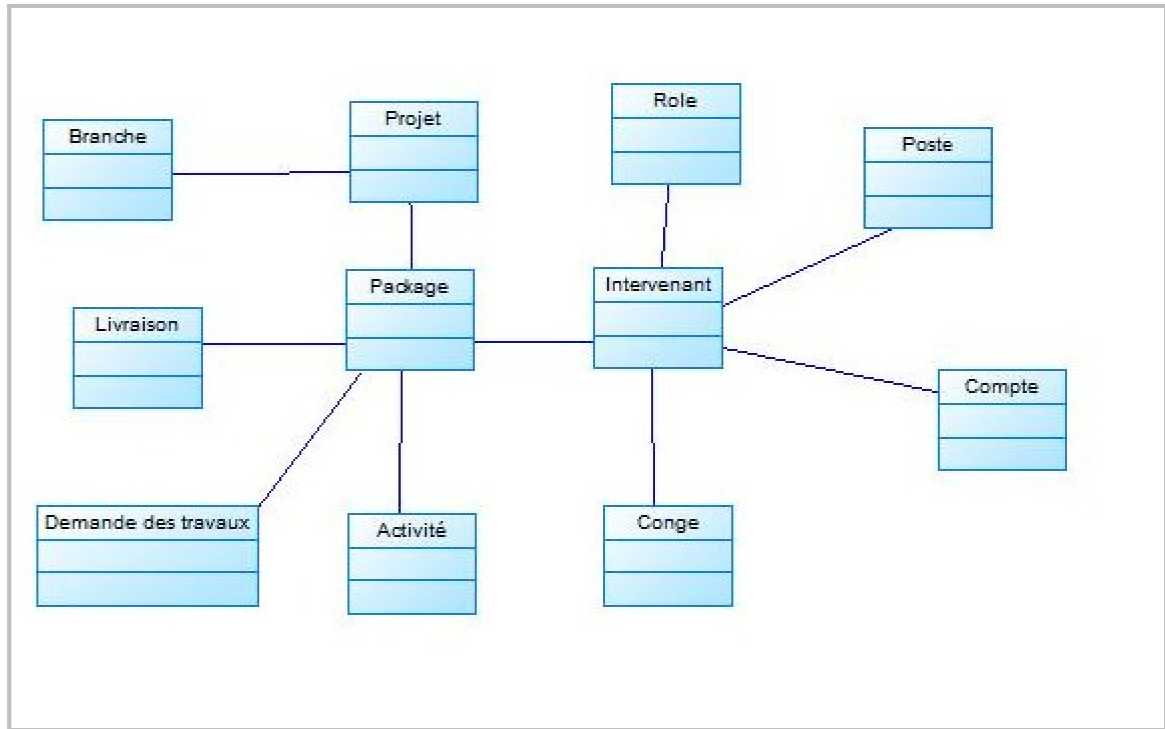


Figure 25: Diagramme des classes candidates

Chapitre II : Capture des besoins techniques et analyse



Dans ce chapitre, je vais définir les modèles de spécification matérielle et logicielle, le modèle de déploiement et finalement l'organisation du système en couches .



II.1. Capture des besoins techniques

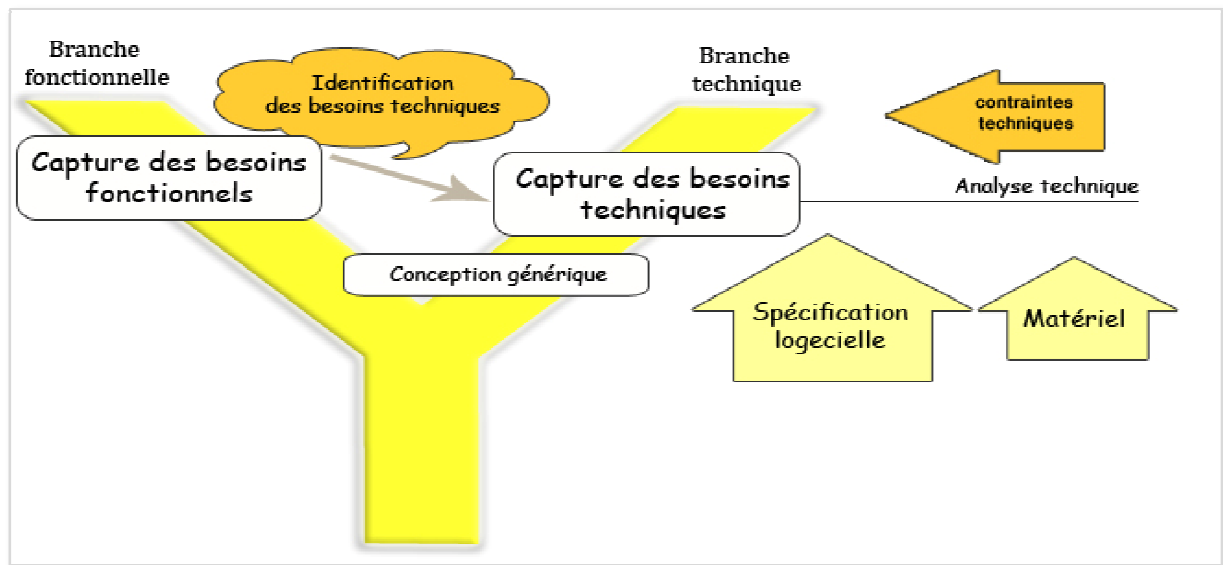


Figure 26: Situation de la capture des besoins techniques dans 2TUP

II.1.1 Organisation du modèle de spécification logicielle

Le modèle de spécification logicielle obtenu précédemment est plus au moins sommaire car si on prend un cas d'utilisation technique on trouve qu'il concerne différentes responsabilités de traitement qui vont de l'interface utilisateur à la base de données.

L'organisation du modèle se base sur le modèle en couches qui s'empilent en niveaux pour couvrir des transformations logicielles successives, de sorte que la couche d'un niveau ne puisse utiliser que les services des couches des niveaux inférieurs.

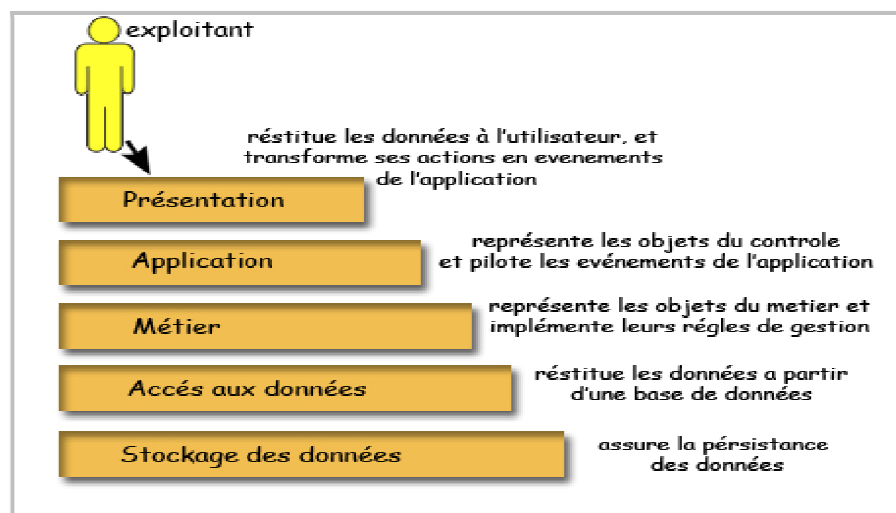


Figure 27: Architecture en couches

Dans le modèle UML, les couches logicielles correspondent à des packages. Ces packages contiennent des cas d'utilisation techniques.

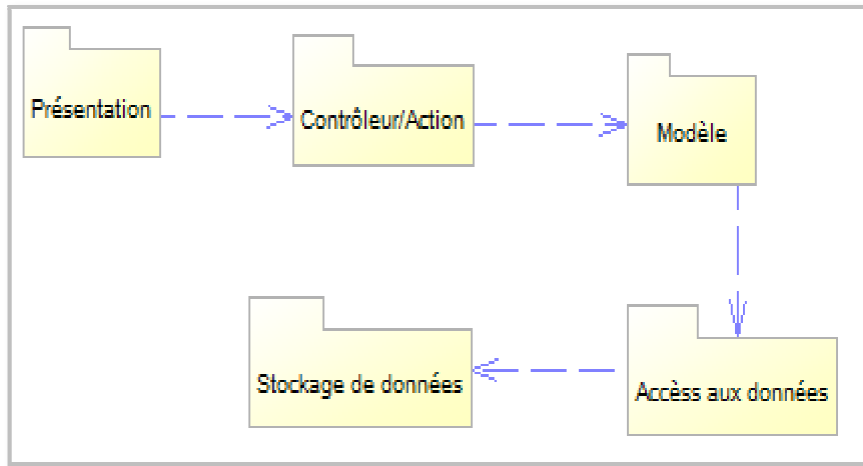


Figure 28: diagramme de packages

II.1.2. Définition des concepts techniques

Dans cette étape il faut exprimer pour chaque package un vocabulaire spécifique correspondant à la description du point de vue technique qu'il représente. Le tableau ci-après fournit un aperçu des différents concepts utilisés au niveau de chacun des package.

Couche logicielle	Concept manipulé
Présentation	JSF 2.0 ,PrimeFaces3.2
Application	Managed Bean
Métier	EJB3
Accès aux données	,Eclipcelink,JPA2
Stockage de données	Derby

Tableau 12: Concepts techniques du système

II.2.Analyse

Ce chapitre traite l'analyse objet du système. Nous verrons en particulier comment j'ai utilisé la notion de package pour définir des catégories de classes d'analyse, les principales constructions du diagramme de classes UML et finalement les différents scénarios.

II.2.1.Découpage en catégories

Une approche objet offre à l'analyste la possibilité de consolider la définition des concepts, en termes de représentation à la fois structurelle et comportementale, puis de réutiliser ces définitions. Le découpage initial fonctionnel doit donc être remis en cause si l'on veut profiter des bénéfices de l'approche objet ; le découpage en catégories devient alors la base du modèle structurel d'analyse.

Ce découpage initial doit se fonder sur l'ensemble des classes candidates identifiées durant la phase précédente, mais également sur deux principes fondamentaux : *cohérence* et *indépendance*

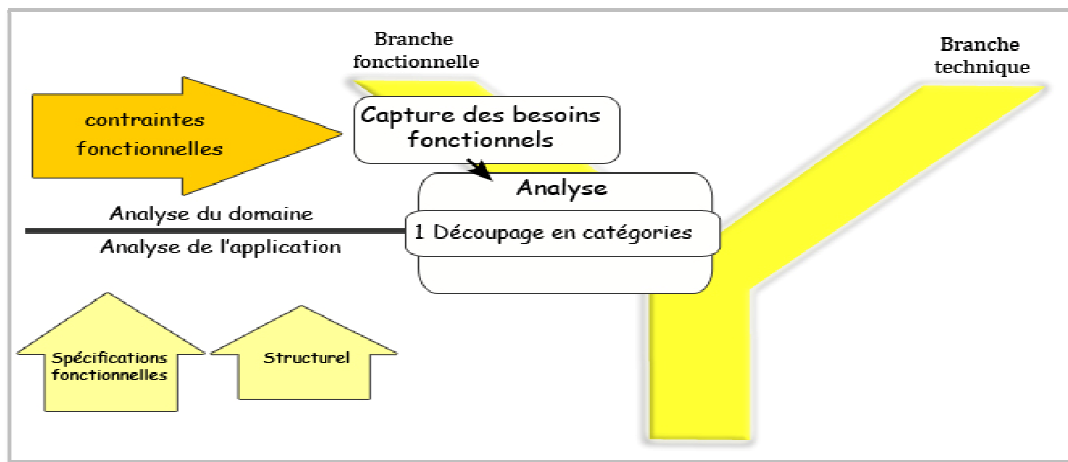


Figure 29: Situation du découpage en catégories dans 2TUP

Découpage en catégories des classes du système

Cette partie traite le démarrage de l'analyse objet du système à réaliser. Elle consiste à découper le système en packages de classes appartenant à des catégories.

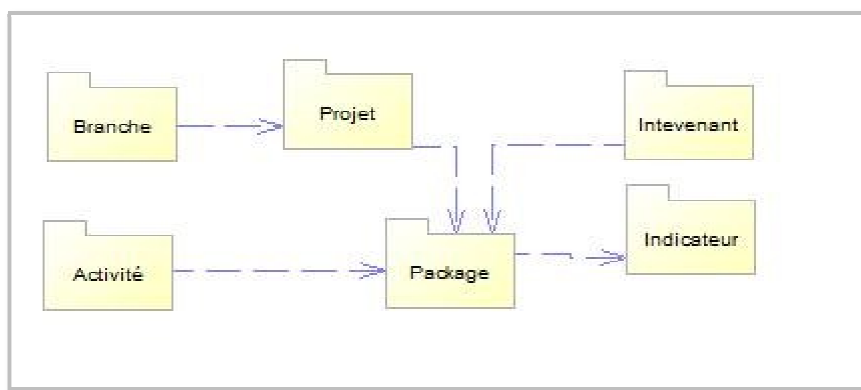


Figure 30 :Diagramme de packages

Liste des classes par package ou catégorie

Catégorie	Classes de la catégorie
-----------	-------------------------

Package	Package, livraison, demande des travaux,
Activité	Activité, type d'activité
Branche	Branche
Projet	Projet
Intervenant	Intervenant , rôle, poste, congé
Indicateur	indicateur

Tableau 13: classes du système par catégories

Après avoir déterminé les packages du système il fallait élaborer un diagramme de classes pour chaque catégorie pour bien illustrer la structure interne de chacune.

Diagrammes de classes par catégorie

Catégorie Package

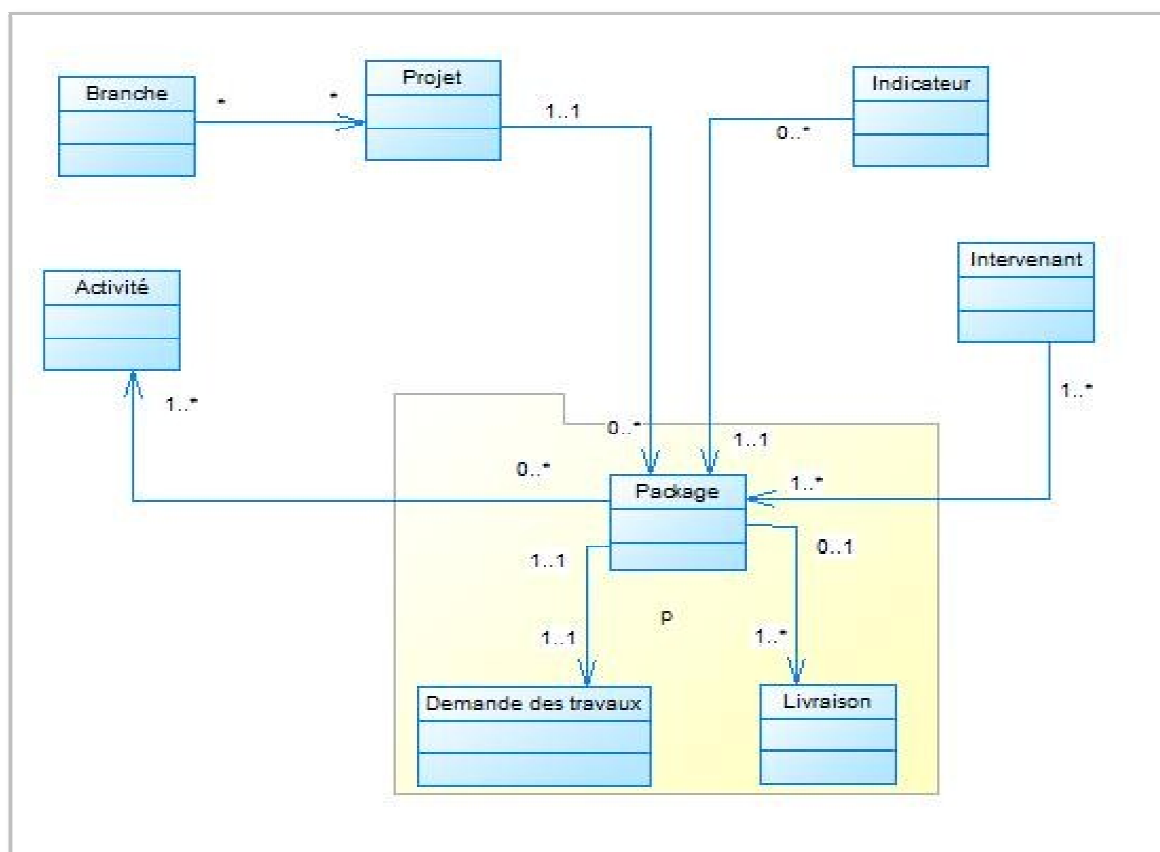


Figure 31 : Diagramme de classes de la Catégorie Package

Catégorie Activité

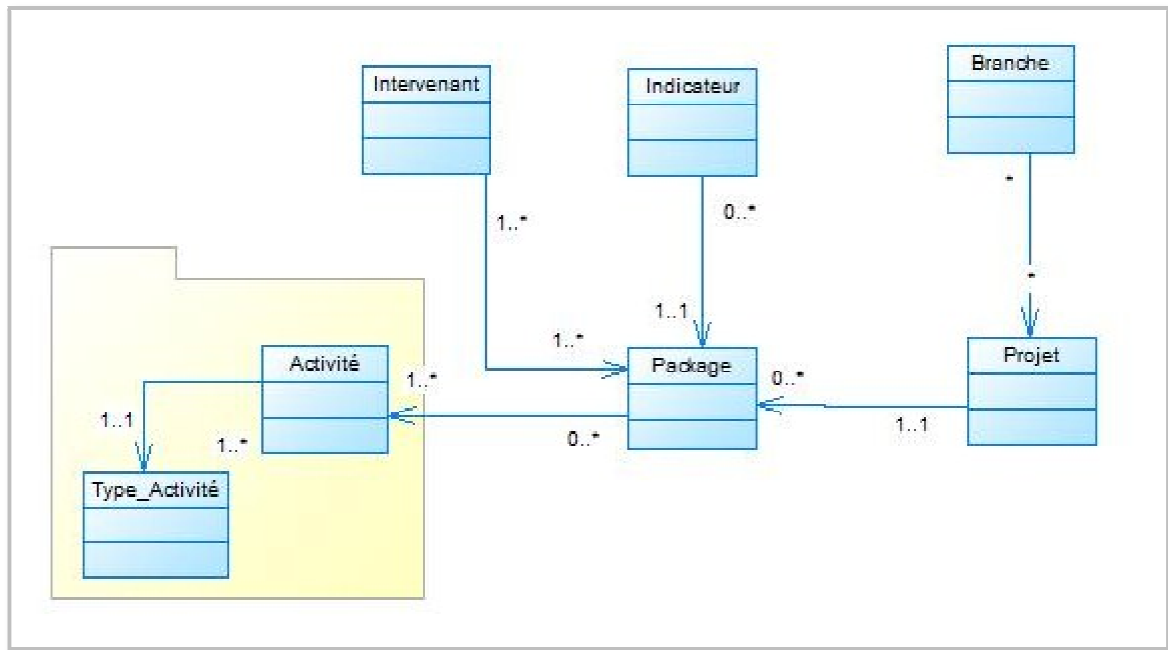


Figure 32:Diagramme de classes de la Catégorie Activité

Catégorie Branche

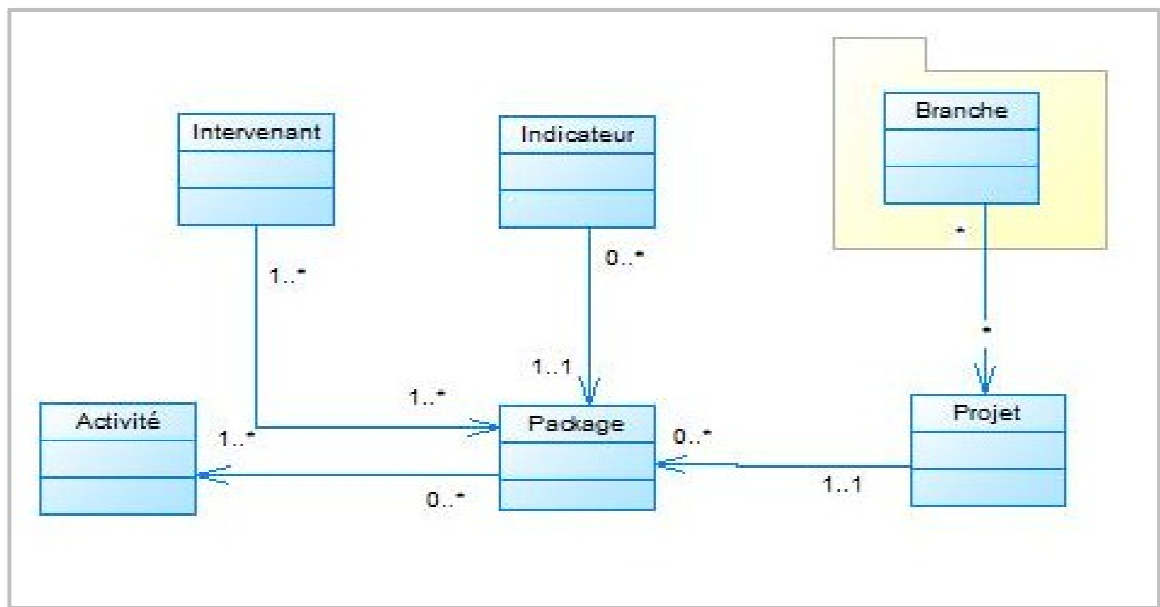


Figure 33:Diagramme de classes de la Catégorie Branche

Catégorie Projet

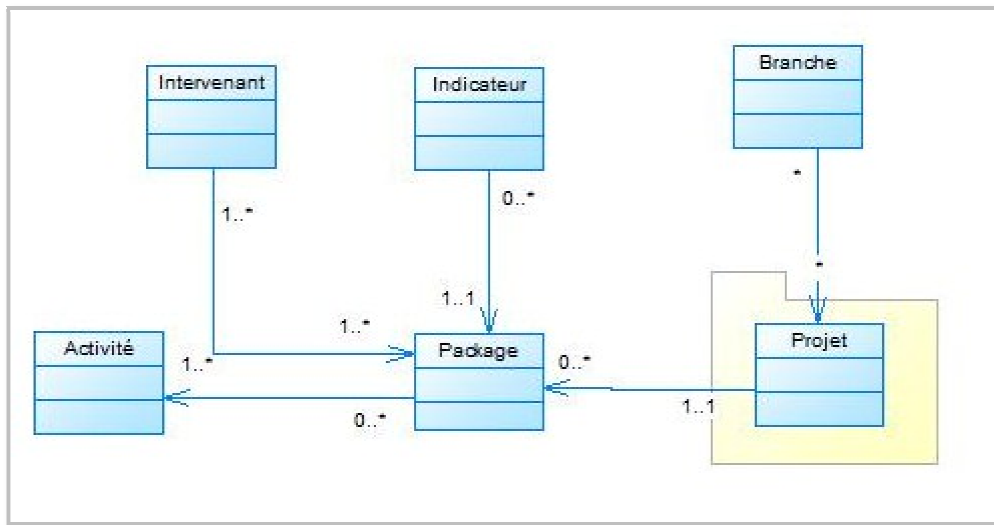


Figure 34:Diagramme de classes de la Catégorie projet

Catégorie Intervenant

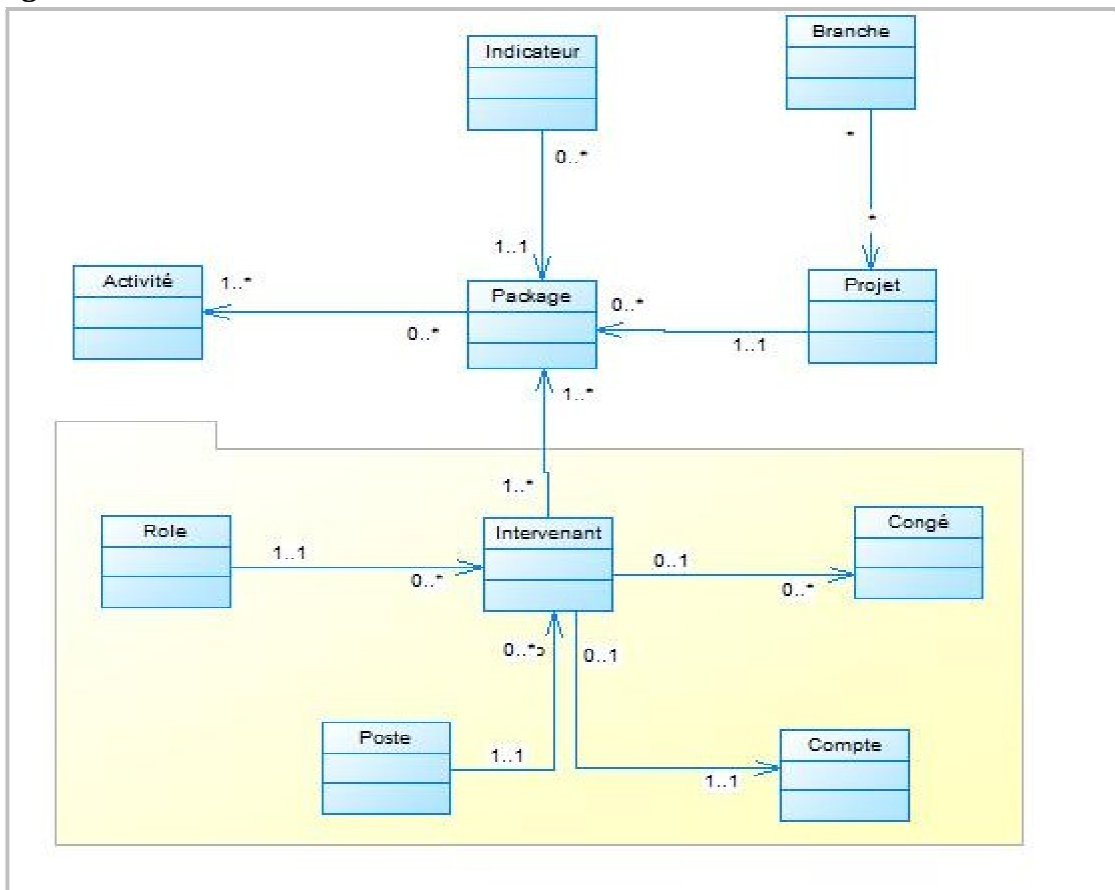


Figure 35 :Diagramme de classes de la Catégorie intervenant

Catégorie Indicateur

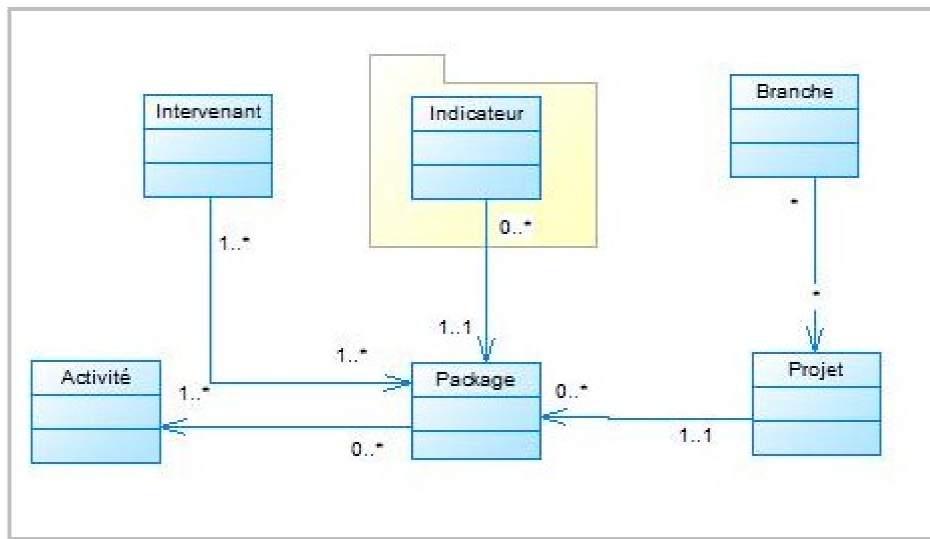


Figure 36: Diagramme de classes de la Catégorie indicateur

II.2.2. Développement du modèle statique

Ce chapitre va nous permettre d'illustrer les principales constructions du diagramme de classes UML durant l'étape d'analyse.

Le diagramme de classes a toujours été le diagramme le plus important dans toutes les méthodes orientées objet. C'est également celui qui contient la plus grande gamme de notations et de variantes.

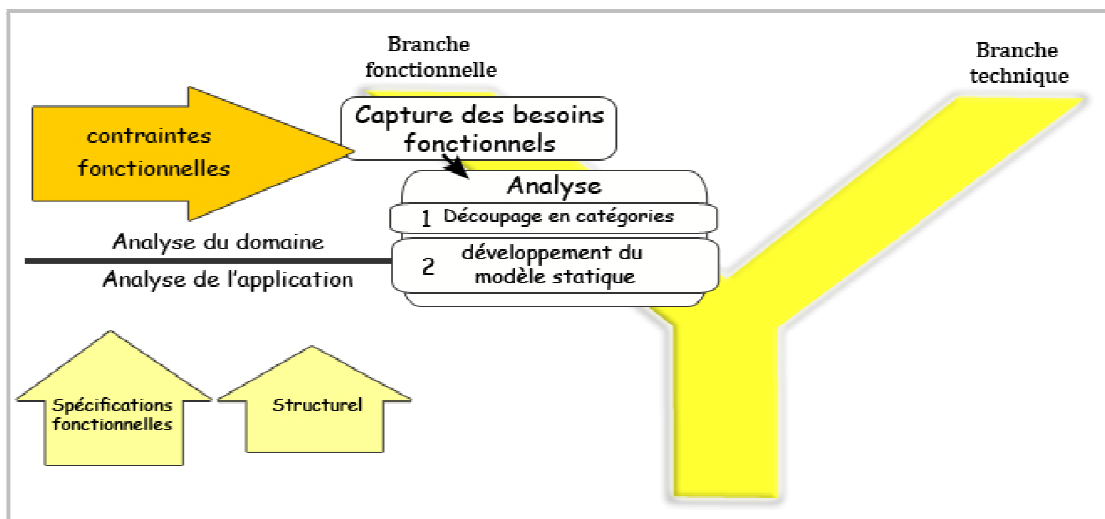


Figure 37: Situation du modèle statique dans 2TUP

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y et succède au découpage en catégories. Les classes

identifiées lors de l'étude des cas d'utilisation, puis réparties dans les catégories, sont simplement des classes candidates pour l'analyse objet. Je vais les évaluer en détail, en éliminant certaines, ou au contraire en ajoutant d'autres.

Affinement des associations

Les associations identifiées jusqu'à présent ne forment qu'une ébauche de structure statique. Il convient dans cette étape de les valider, les préciser.

Affinement des attributs

Dans un modèle d'analyse, seules les propriétés simples des classes que le système doit mémoriser et utiliser, sont conservées comme attributs. Il s'agit donc de raffiner les classes en leur rajoutant ou en leur enlevant des attributs.

Optimisation des classes avec la généralisation

Il s'agit de découvrir les classes possédant des caractéristiques communes: attributs, associations, opérations.

Ajout des opérations

Une opération représente un service, un traitement qui peut être demandé à n'importe quel objet de la classe. Une opération est l'abstraction de ce que vous pouvez réaliser sur un objet, et elle est partagée par tous les objets de la classe.

Souvent, invoquer une opération sur un objet modifie son état ou les valeurs de certains attributs, ou encore l'existence de certains liens avec d'autres objets.

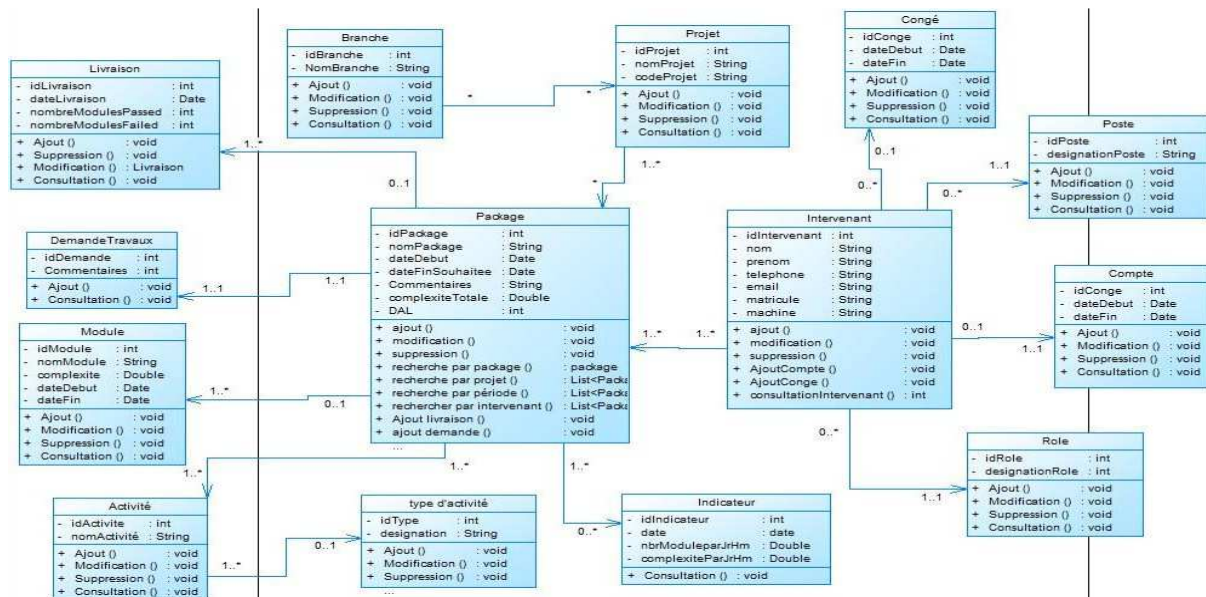


Figure 38: Diagramme de classe avec les opérations et les attributs.

II.2.3. Développement du modèle Dynamique

Le développement du modèle dynamique précède l'étape de conception préliminaire. Il s'agit d'une activité itérative, fortement couplée avec la modélisation statique.

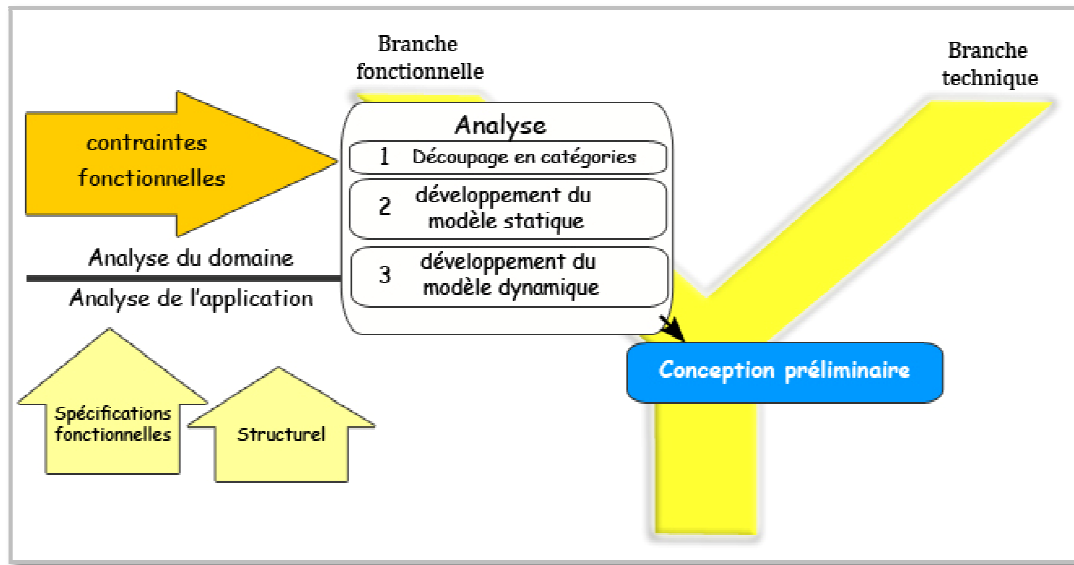


Figure 39: Situation du développement du modèle dynamique dans 2TUP

Diagrammes de séquence

Un cas d'utilisation décrit un ensemble de scénarios. Un scénario décrit une exécution particulière d'un cas d'utilisation du début jusqu'à la fin. Il correspond à une sélection d'enchaînements du cas d'utilisation. Il est clair que la combinatoire des enchaînements fait exploser le nombre de scénarios potentiels, je ne peux pas donc pas tous les décrire. Alors les diagrammes de séquence que j'ai réalisés sont les suivants :

Authentification :

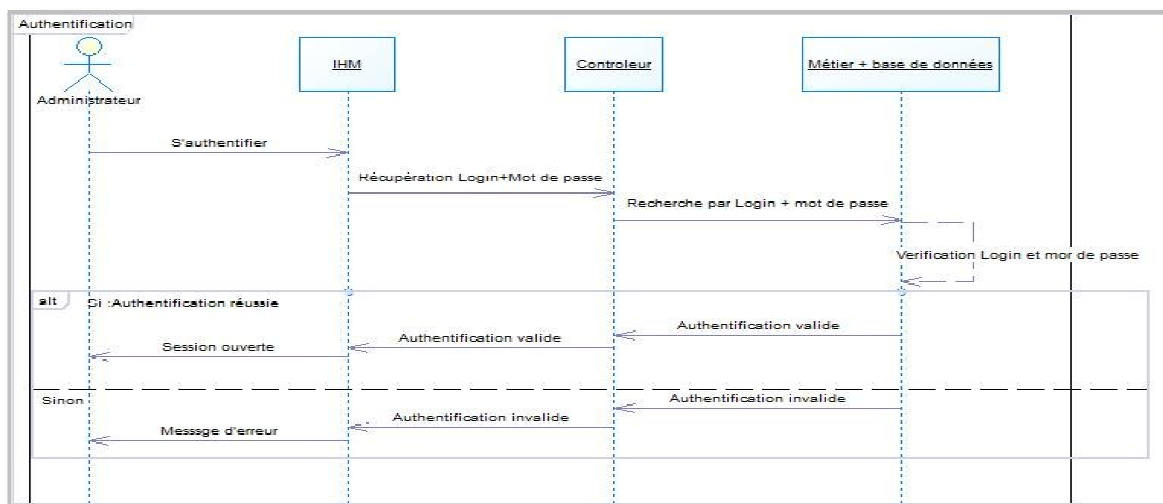


Figure 40: Diagramme de séquence de la partie authentification
Consultation objet dans le cas d'administrateur :

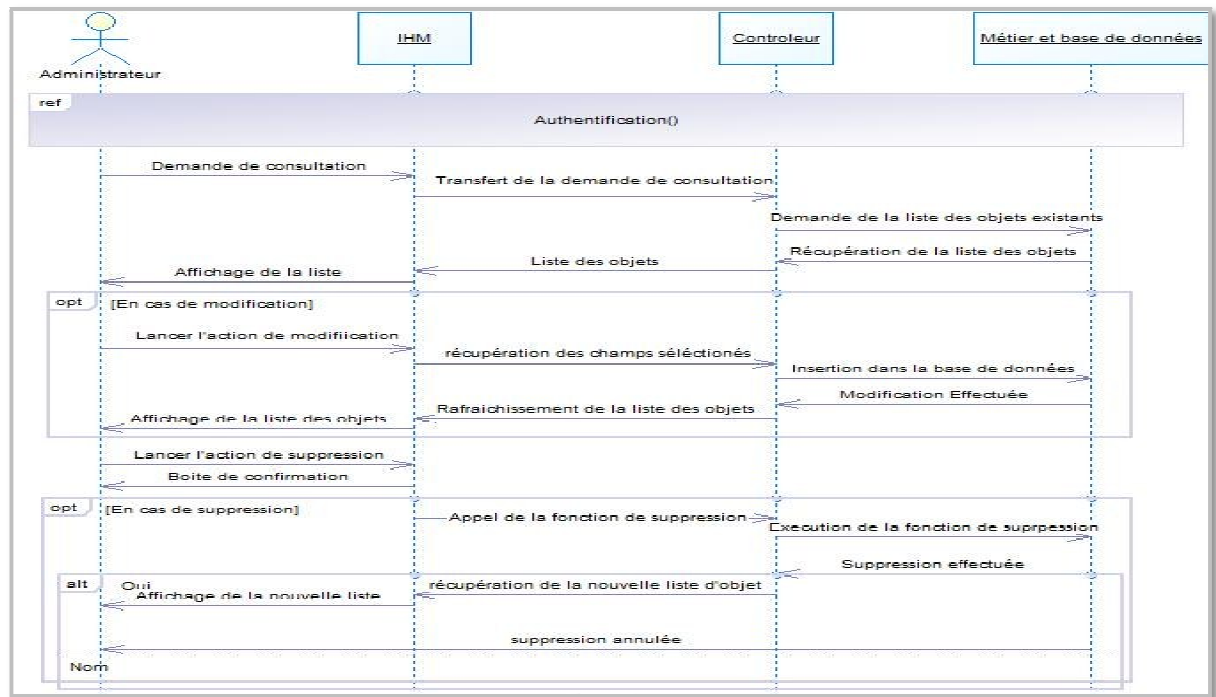


Figure 41: Diagramme de séquence de la partie consultation pour un administrateur

Consultation objet dans le cas d'un simple utilisateur :

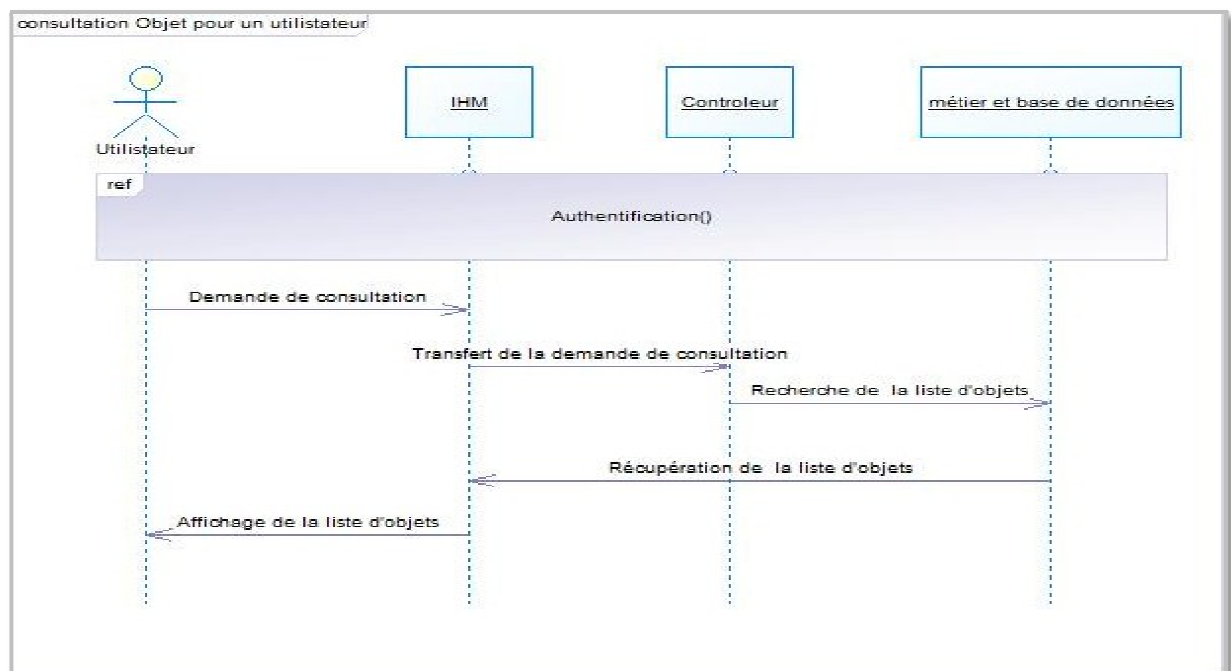


Figure 42: Diagramme de séquence de la partie consultation pour un utilisateur

Ajout d'un objet :

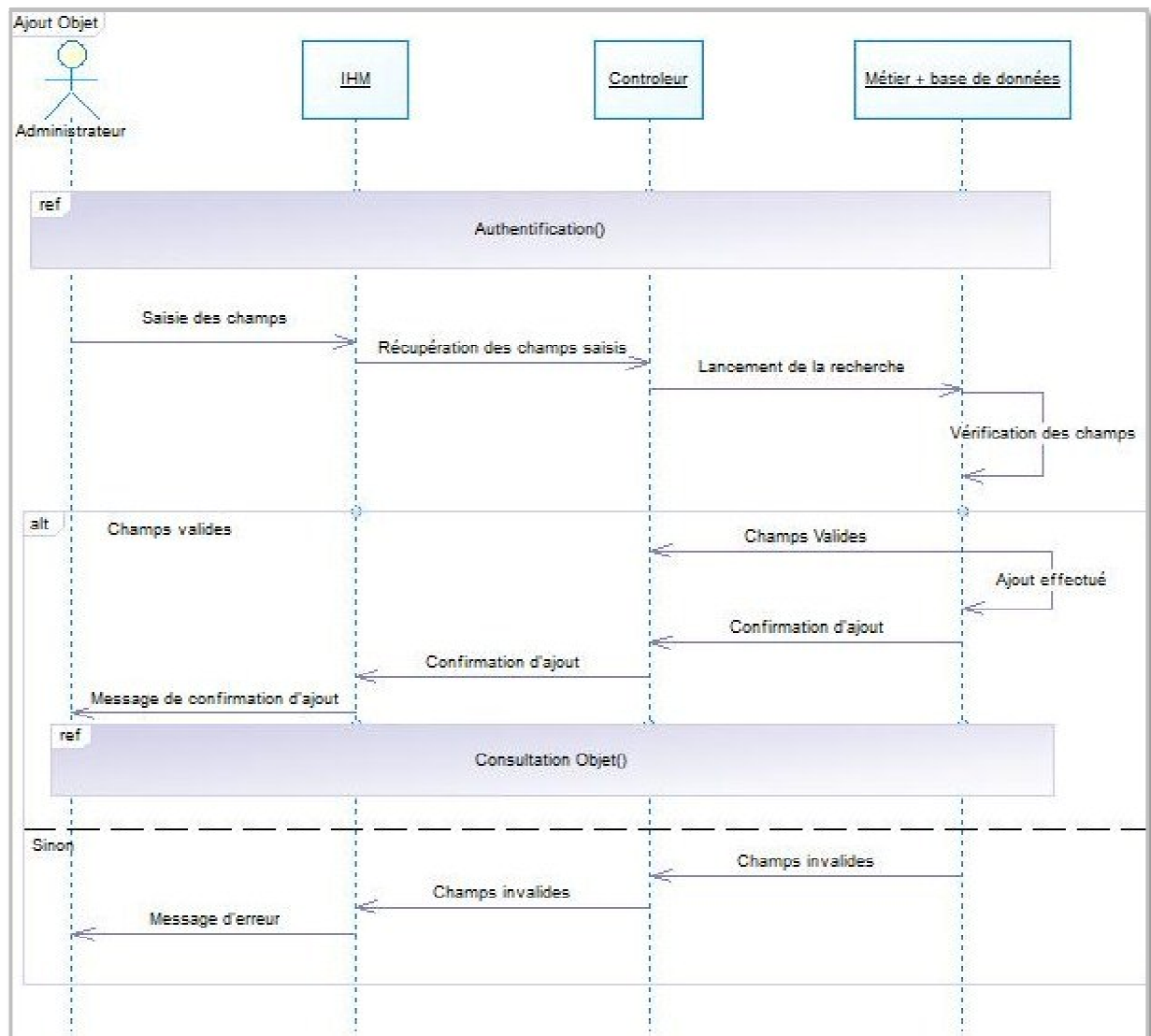


Figure 43: Diagramme de séquence de la partie ajout objet

Consultation intervenant dans le cas d'administrateur :

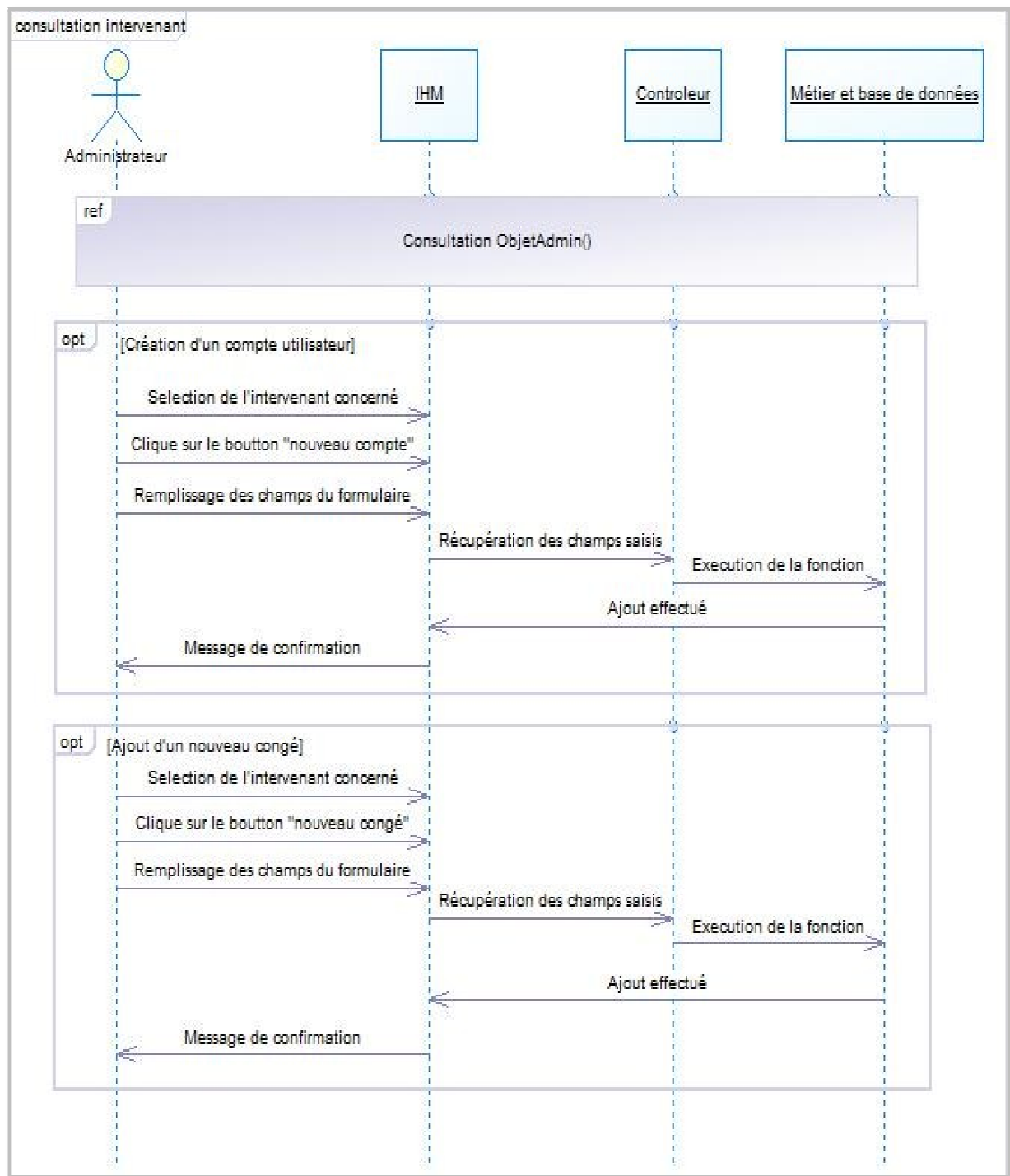


Figure 44: Diagramme de séquence de la partie consultation intervenant

Consultation package dans le cas d'administrateur :

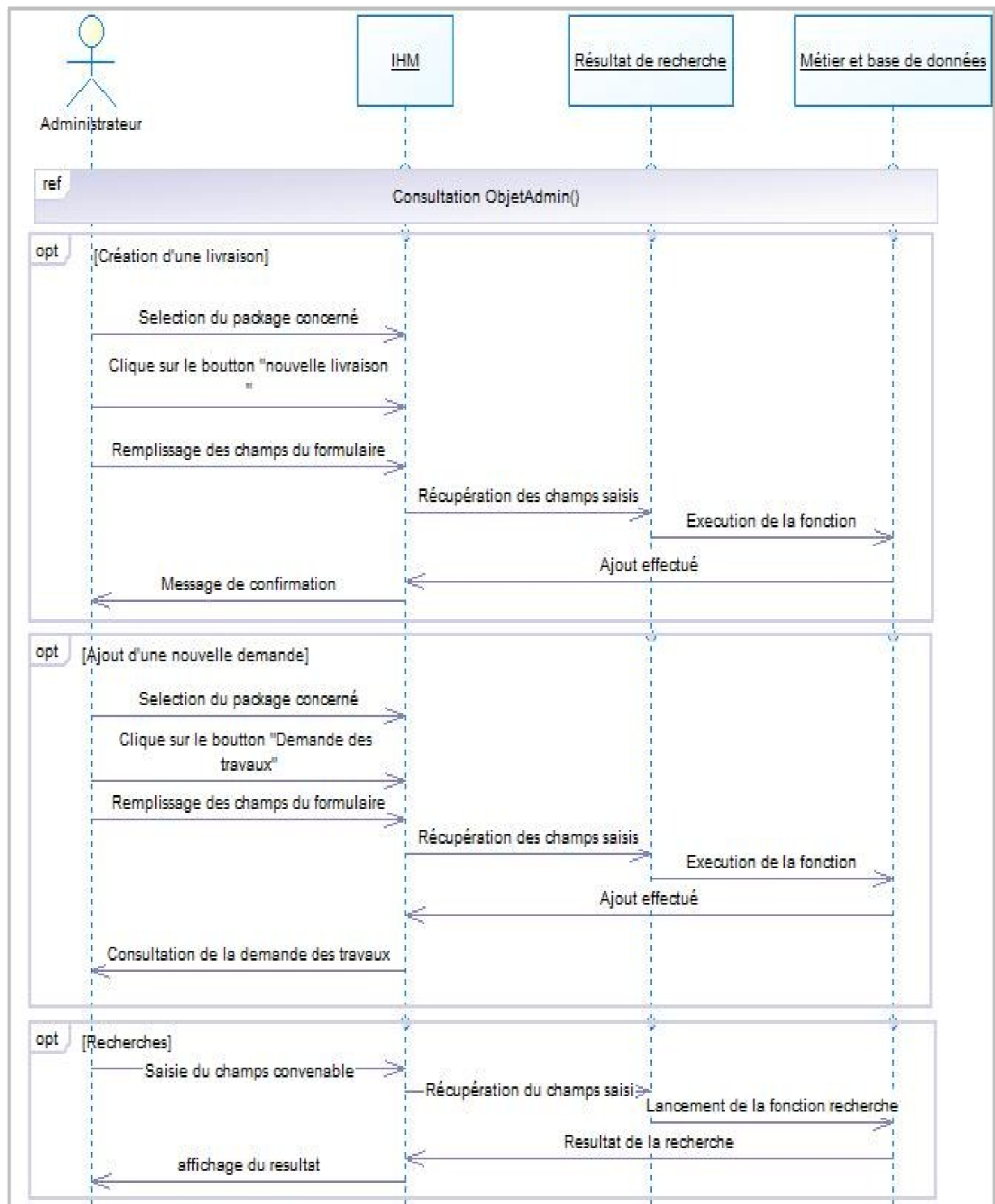


Figure 45: Diagramme de séquence de la partie consultation package

Chapitre III : Conception générique, préliminaire et détaillée



Dans ce chapitre je vais détailler la conception générique, préliminaire et détaillée.

Pour la première conception, elle est qualifiée de générique car elle est entièrement indépendante des aspects fonctionnels spécifiés en branche gauche. La conception générique reste donc une activité de la branche droite du processus 2TUP.

Dans la conception préliminaire, je vais élaborer et expliquer le diagramme de déploiement ainsi que le diagramme d'exploitation.

Enfin, dans la conception détaillée, je vais parler du microprocessus de la conception ainsi que la architecture de l'application.



III.1. Conception générique

La conception générique consiste à développer la solution qui répond aux spécifications techniques que nous avons déjà présentées. Cette conception est qualifiée de générique car elle est entièrement indépendante des aspects fonctionnels spécifiés en branche gauche. La conception générique reste donc une activité de la branche droite du processus 2TUP.

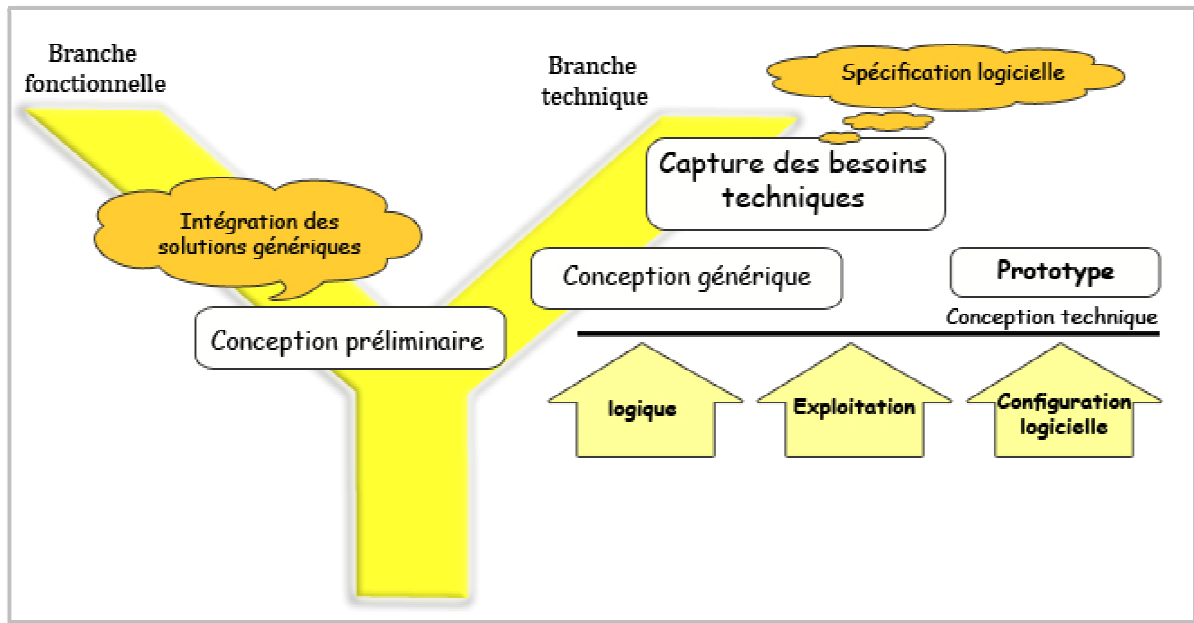


Figure 46: Situation de la conception générique dans 2TUP

La conception générique développe le squelette technique d'un projet. Si ce squelette est robuste et bien conçu, il pourra soutenir toutes les évolutions fonctionnelles possibles. Si, au contraire, il est mal conçu et qu'il nécessite par la suite de nombreuses retouches, la reprise des erreurs aura des répercussions pouvant se révéler très coûteuses en question du temps pour la suite du projet.

III.1.1. Classes et frameworks techniques

On constate cependant généralement qu'une classe technique fonctionne rarement seule, c'est pourquoi le concept-clé de la conception générique est le Framework technique.

Framework technique

Un Framework est un réseau de classes qui collaborent à la réalisation d'une responsabilité qui dépasse celle de chacune des classes qui y participent.

Un Framework peut être abstrait ou concret. Dans le premier cas, il est constitué d'interfaces. La réutilisation du Framework consiste alors à implémenter ces interfaces, à charge pour le développeur de comprendre et de respecter le domaine de responsabilité technique qui leur échoit. Un Framework abstrait structure seulement le modèle de configuration logicielle. Il est composé de classes que l'on peut directement réutiliser dans un projet. Un Framework concret détermine à la fois le modèle de configuration logicielle et le modèle d'exploitation. En réalité, un Framework n'est pas forcément abstrait ou concret, il est plus souvent une combinaison d'interfaces à implémenter et de classes à réutiliser. Au niveau du modèle d'exploitation, il est livré sous la forme d'un package, d'un fichier JAR, WAR, EAR ou plus largement d'une bibliothèque. Dans le modèle d'exploitation, un Framework peut donner lieu à des composants distribués ou à des librairies partagées.

Frameworks techniques de l'application

Ce projet fait appel à JSF 2.0 qui est un framework de la couche présentation.

III.1.2. Organisation des frameworks techniques

L'organisation du modèle logique reprend les couches logicielles. À chaque couche correspond un ou plusieurs frameworks techniques, en partie abstrait, qui définit des interfaces de réalisation des responsabilités logicielles :

- la couche présentation définit les classes, les interfaces et les design patterns nécessaires pour réaliser la gestion des objets et des interfaces en vue de leur utilisation par les utilisateurs, notamment on a fait appel au framework techniques JSF 2.0 ,avec l'utilisation de la librairie PrimeFaces 3.2.
- la couche applicative définit ces mêmes éléments pour rafraîchir les vues, charger les modèles de fonctionnement et contrôler les commandes des clients via l'utilisation des ManagedBean.
- la couche métier définit les éléments permettant d'identifier les objets métier et de les gérer. En utilisant les EJB3.
- la couche d'accès aux données définit les mécanismes de chargement, de sauvegarde, de mise à jour et de recherche des objets persistants, cette dernière utilise l'API : JPA 2.

III.1.3 Développement du prototype technique

J'ai conçu tous les Framework techniques qui me permettront de répondre aux spécificités techniques de système. Ainsi l'architecture technique réalisée garantit l'intégrité conceptuelle de tous futurs développements. Je contribue ainsi non seulement à une bonne partie du travail de conception, mais apportons également les meilleurs savoir-faire à reproduire pour le reste du projet.

C'est pour cela que j'ai met en place un prototype technique qui va éliminer le risque technique et qui va égerment détailler les point suivants:

- le mécanisme CRUD des objets (Create, Retrieve, Update, Delete) depuis la présentation jusqu'à la base de données
- les transformations d'objets entre les couches
- les mécanismes de présentation sur le poste client.
- Le prototype valide et termine éventuellement l'étape de conception générique. Nous sommes maintenant arrivés au terme de ce chapitre et allons en récapituler les phases de réalisation.

III.2. Conception préliminaire

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle en représente le cœur. C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques. En conséquence, plusieurs activités doivent coexister :

- Passer de l'analyse objet à la conception,
- Intégrer les fonctions métier et applicatives du système dans l'architecture technique,
- Adapter la conception générique aux spécifications fournies par l'analyse. La conception préliminaire est avant tout affaire d'organisation ; elle s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse, mais également sur les *frameworks* de la conception technique. Elle se termine lorsque la conception est organisée suivant :
 - son déploiement cible,
 - son modèle d'exploitation,
 - son modèle logique.

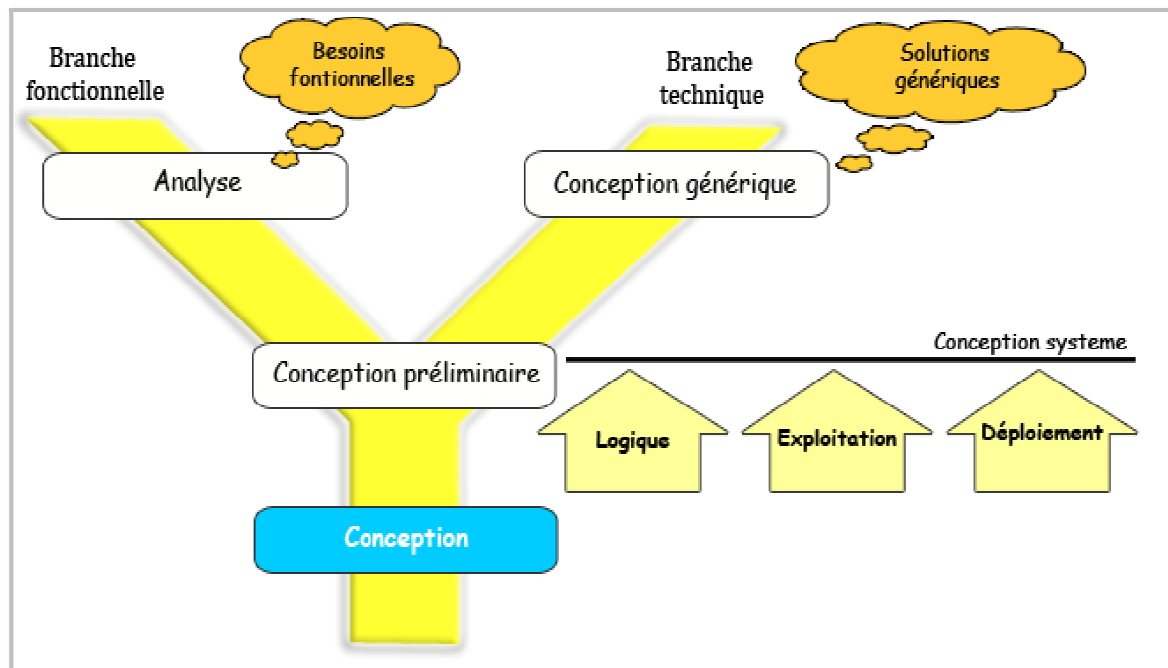


Figure 47: Situation de la conception générique dans 2TUP

Le premier niveau de conception d'un système est son déploiement car c'est généralement l'organisation des environnements de travail sur un réseau qui est la plus immédiatement définie.

III.2.1. Développement du modèle de déploiement

Le déploiement d'une solution client/serveur se construit sur la définition des postes de travail.

Poste du travail : le poste de travail représente un ou plusieurs acteurs pouvant être localisés sur une machine d'un type particulier et remplissant une fonction identifiée dans l'entreprise. Le poste de travail ne représente pas forcément une machine physique, mais peut consister en plusieurs machines, à condition qu'elles donnent lieu au même type de déploiement.

Le déploiement du système est le suivant:

- l'environnement de travail « Client » est déployé sur des machines prenant en charge l'exécution d'une machine virtuelle Java et tournent sur n'importe quel système d'exploitation.
- La base des données du système est déployée sur serveur installé sur une des machines de la société.

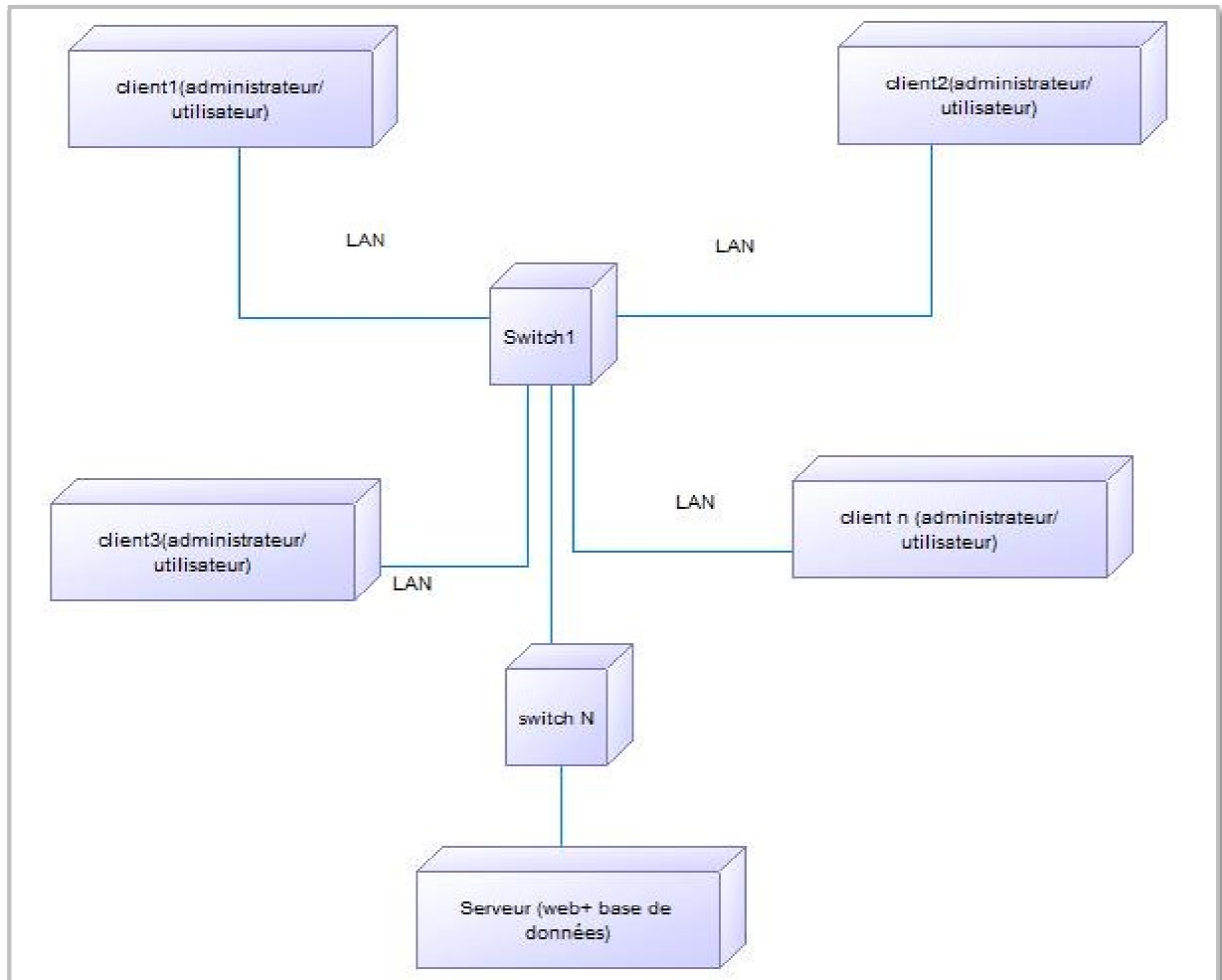
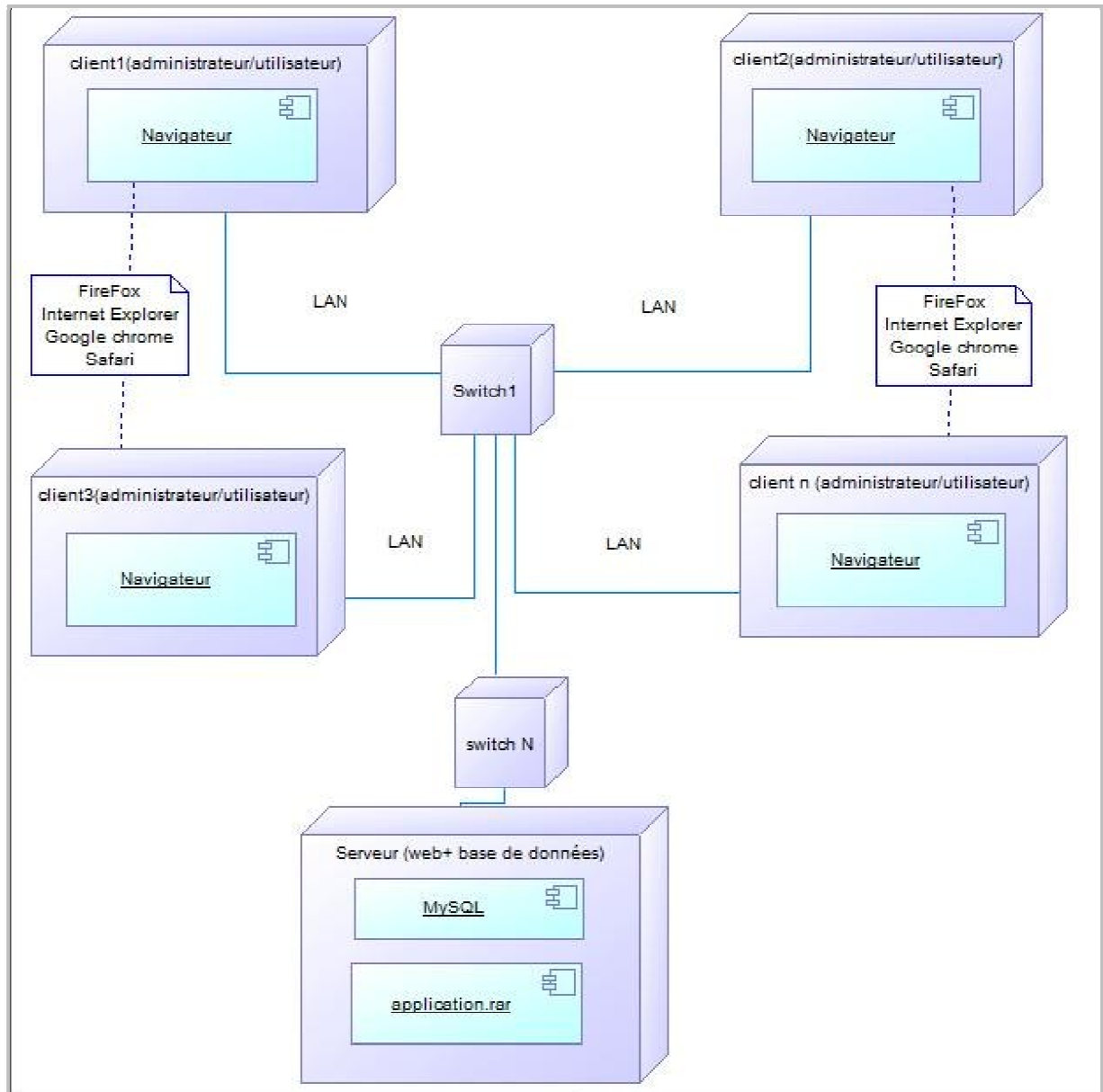


Figure 48: Diagramme de déploiement du système

III.2.2 Développement du modèle d'exploitation

L'élaboration d'une architecture d'exploitation a déjà commencé en conception générique. À partir du déploiement, il est possible de la compléter en fonction des machines et des postes de travail, tout en intégrant les besoins exprimés en analyse.

Le développement du modèle d'exploitation consiste à compléter le modèle de déploiement en déterminant les différentes applications qui vont s'installer sur les postes du travail et serveurs, en plus les instances de la base de données.

Définition des applications :**Figure 49: Model d'exploitation – définition des applications****III.2.3. Énumération des interfaces utilisateur**

Dans le cadre d'entreprises dont la culture de spécification est fortement axée sur la définition d'écrans, il est fort judicieux d'accompagner le développement 2TUP de la définition des IHM.

À l'étape de capture des besoins fonctionnels, ceux d'IHM peuvent se transformer en maquettes pour chaque cas d'utilisation. Leur présentation aux utilisateurs pourra susciter des remarques et avoir des répercussions sur les règles de gestion métier.

À l'étape d'analyse, au niveau de l'analyse de l'application plus particulièrement, les messages produits ou reçus par les acteurs peuvent être attachés aux vues déjà définies. Cette pratique permettra de mieux fixer le rôle fonctionnel de chaque vue et d'améliorer éventuellement leur ergonomie.

À l'étape de conception préliminaire, les vues de la maquette sont formalisées et réparties sur les différentes applications du modèle d'exploitation. L'identification des composants d'IHM et la réutilisation au niveau de la couche de présentation s'en trouveront affinées. Leur présentation auprès des utilisateurs permettra enfin d'améliorer l'ergonomie de fonctionnement.

Le tableau ci-dessous décrit les interfaces exploitées par l'application et donne une idée plus précise de l'environnement de travail du client :

Vue IHM	Description
Ajout d'un nouveau Rôle	Permet à l'administrateur de créer un nouveau rôle
Ajout d'un nouveau Poste	Permet à l'administrateur de créer un nouveau poste
Ajout d'un nouveau intervenant	Permet à l'administrateur de créer un nouveau intervenant
Ajout d'une nouvelle branche	Permet à l'administrateur de créer une nouvelle branche
Ajout d'un nouveau projet	Permet à l'administrateur de créer un nouveau projet
Ajout d'un nouveau package	Permet à l'administrateur de créer un nouveau package
Ajout d'une nouvelle activité	Permet à l'administrateur de créer une nouvelle activité
Ajout d'un nouveau type d'activité	Permet à l'administrateur de créer un nouveau type d'activité
Consultation rôle	Permet à l'administrateur de : consulter la liste des rôles ▪ modifier un rôle ▪ supprimer un rôle
Consultation poste	Permet à l'administrateur et à l'utilisateur de consulter la liste des postes. Permet à l'administrateur de : ▪ modifier un poste ▪ supprimer un poste
Consultation Intervenent	Permet à l'administrateur et à l'utilisateur de consulter la liste des intervenants. Permet à l'administrateur de : ▪ modifier un intervenant ▪ supprimer un intervenant ▪ créer un nouveau compte ▪ créer un nouveau congé

Consultation activité	Permet à l'administrateur et à l'utilisateur de consulter la liste des activités. Permet à l'administrateur de : ▪ modifier uneactivité ▪ supprimeruneactivité
Consultation branche	Permet à l'administrateur et à l'utilisateur de consulter la liste des branches. Permet à l'administrateur de : ▪ modifier unebranche ▪ supprimerunebranche
Consultation projet	Permet à l'administrateur et à l'utilisateur de consulter la liste des projets. Permet à l'administrateur de : ▪ Modifier un projet ▪ Supprimer un projet
Consultation package	Permet à l'administrateur et à l'utilisateur de : ▪ Consulter la liste des packages. ▪ Rechercher un package par nom/projet/période/intervenant ▪ Créerune nouvelle livraison ▪ Créerune nouvelle demande ▪ Permet à l'administrateur de : ▪ modifier un package ▪ supprimer un package
Consultation livraison	Permet à l'administrateur et à l'utilisateur de consulter la liste des livraisons. Permet à l'administrateur de : ▪ modifier unelivraison ▪ supprimerunelivraison
Consultation compte	Permet à l'administrateur et à l'utilisateur de consulter la liste des comptes. Permet à l'administrateur de : ▪ modifier un compte ▪ supprimer un compte
Consultation congé	Permet à l'administrateur et à l'utilisateur de consulter la liste des congés. Permet à l'administrateur de : ▪ modifier un congé ▪ supprimer un congé
Consultation demande	Permet à l'administrateur et à l'utilisateur de consulter la demande.

Tableau 14: Liste des IHM du système

III.3 Conception détaillée

L'analyse du système s'appuie sur les besoins et vis à comprendre les problèmes métiers qui sont posés, en faisant abstraction des considérations techniques et informatiques et de tout aspect de réalisation.

Cette phase consiste à déterminer les éléments qui interviennent dans le fonctionnement du système, et de définir leurs structures ainsi que leurs relations. Elle permet aussi d'analyser le système sous deux aspects différents, l'aspect statique et l'aspect dynamique.

La phase d'analyse permet de s'accorder sur « **Ce que doit faire le système** » avant de s'accorder sur « **la manière dont il doit le faire** ».

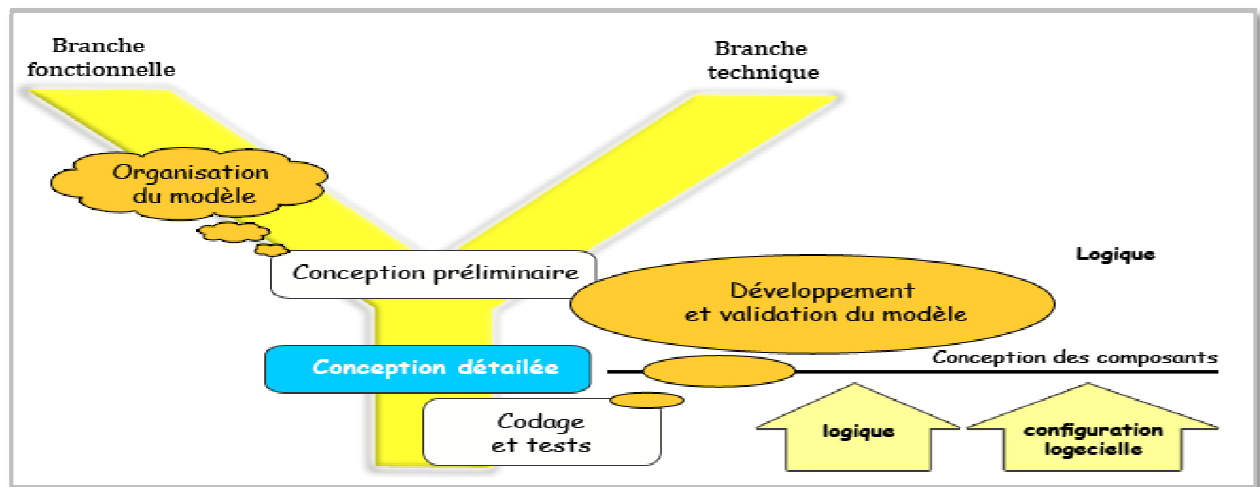


Figure 50: Situation de la conception détaillée dans 2TUP

III.3.1 Le micro-processus de conception logique

Le micro-processus de conception logique concerne la définition des classes à implémenter. C'est donc une activité centrée sur le modèle logique.

Le micro-processus consiste en une itération des cinq activités représentées à la figure suivante :

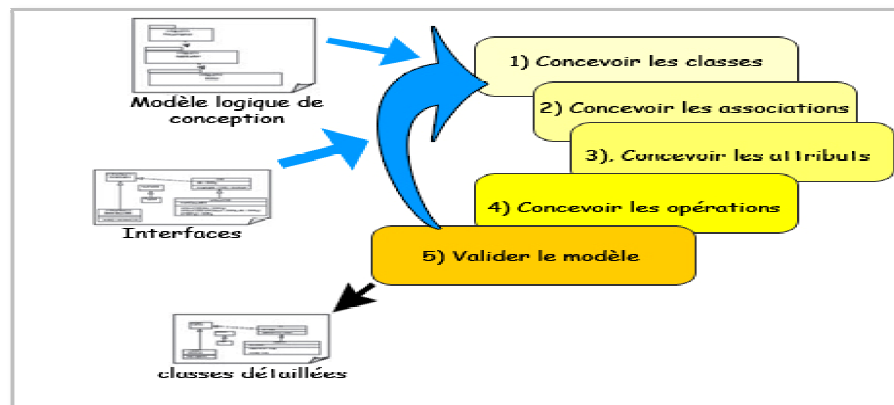


Figure 51: Micro-processus de conception détaillée

Concevoir les classes consiste à transformer des concepts provenant de l'analyse, tels que les métaclasses, en techniques disponibles avec les langages et les outils de développement.

Concevoir les associations définit la façon d'exploiter chaque association et les techniques qui seront employées dans le codage.

Concevoir les attributs nécessite essentiellement d'identifier les structures de données, les itérations et d'autres types complexes permettant de représenter les attributs d'analyse avec le langage utilisé.

Concevoir les opérations permet de déterminer le contenu des méthodes complexes et d'identifier en cascade de nouvelles classes et opérations dans la catégorie.

Valider le modèle constitue la phase de décision du cycle itératif. Sortir de ce cycle signifie que le modèle donne l'image prête à coder de ses composants de configuration logicielle.

IV.3.2. Description du modèle de classes

Conception des attributs de métier

La conception des attributs consiste principalement à définir le type des attributs identifiés en analyse. Bien que la plupart des attributs se satisfont des types de base de Java, certains attributs d'analyse correspondent à une structure de données (classe) qu'il est nécessaire de spécifier.

Concevoir les attributs, c'est également spécifier leur visibilité et leur mode d'accès. Par défaut, un attribut est privé et ce principe reste invariable dans ma conception.

La conception des attributs des classes ne pose guère de problèmes, les méthodes de calcul des attributs dérivés correspondront à la transcription en langage Java des contraintes spécifiées par l'analyste.

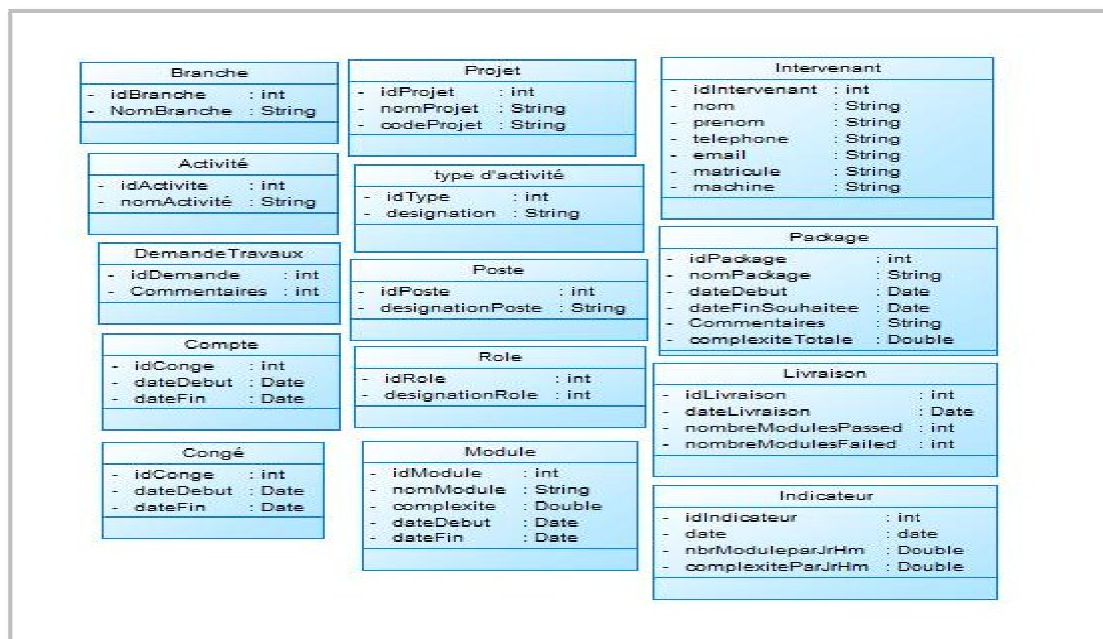


Figure 52: Définition des attributs des classes

Conception des méthodes de métier

La conception des opérations constitue la dernière étape du micro-processus de conception. Le plus gros du travail est certainement fourni lors de cette étape. Il s'agit en effet de donner une image assez précise du contenu des méthodes du projet.

Le développement du modèle dynamique nous a permis de déceler les méthodes qui seront ajoutées aux classes et qui serviront à l'implémentation.

IV.3.3. Conception des couches de l'application

L'architecture que j'ai adoptée dans mon projet est une architecture 3 tiers avec jsf pour la couche présentation, ejb3 pour la couche métier et jpa pour la couche des données.

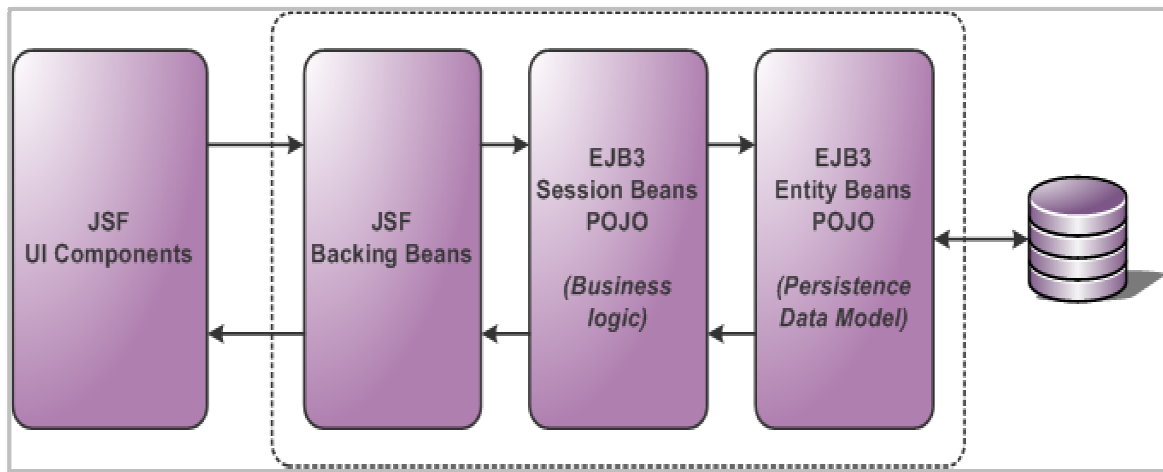


Figure 53: Architecture en couches

La couche présentation est constituée d'un ensemble de composants JSF. Les JSF BackingBeans ou les ManagedBeans sont utilisés pour faire le lien entre la couche présentation et la couche métier, ils permettent ainsi de récupérer les données saisies par les utilisateurs, les valider et les véhiculer à la couche métier. La couche métier est un ensemble de composants EJB session qui encapsulent le logique métier de l'application. Les EJB entités représentent le modèle des données. Ceux-ci seront directement persistés dans la base de données via la couche de persistance qui se charge de la persistance des objets métier manipulés par les services. C'est ici que l'on trouve les fonctionnalités de base qui permettent de créer, rechercher et supprimer des entités métier dans le respect des propriétés transactionnelles. C'est également dans cette couche que la conversion objet/relationnel s'effectue.

Pour plus de visibilité voici les arborescences des différentes couches :

La couche Présentation : L'arborescence des pages web est la suivante :

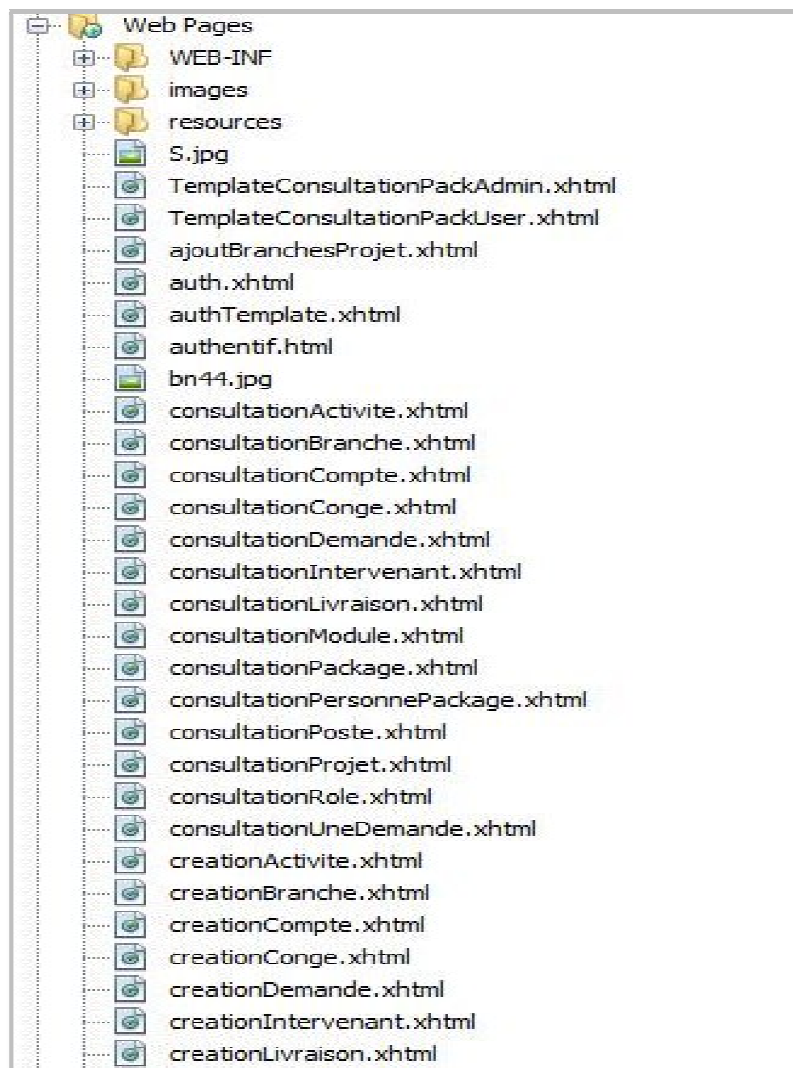


Figure 54: L'arborescence de la couche présentation

Comme je viens de dire à chacune de ces vues est associée une classe ManagedBean permettant de passer les paramètres de l'interface à l'Action. L'arborescence des packages managedBean est la suivante :

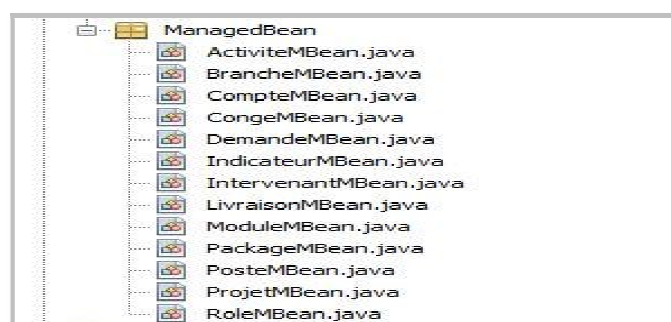


Figure 55: L'arborescence de la couche Contrôleur

La couche métier

L'arborescence des packages métiers est la suivante :

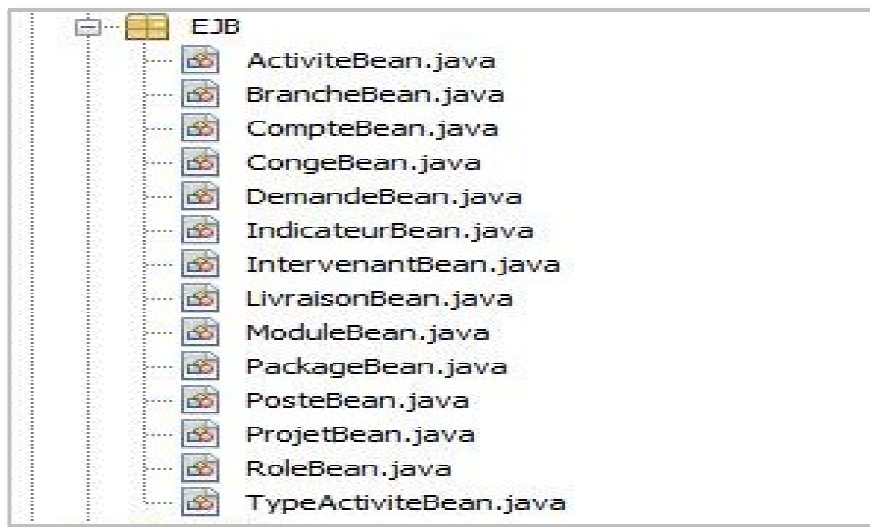


Figure 56: L'arborescence de la couche Métier

La couche persistance

L'arborescence des packages DAO est la suivante :

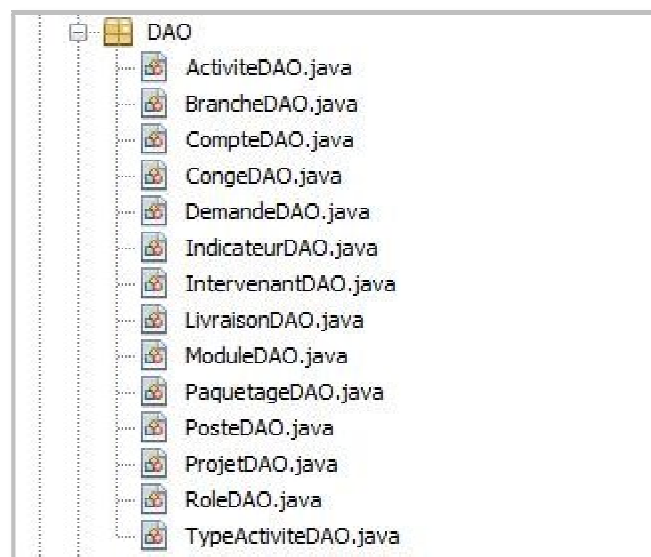


Figure 57: L'arborescence de la couche persistance

Partie III : Mise en œuvre



L'objectif de cette partie est mettre le point sur les technologies utilisées ainsi une visite guidée de l'application.

Le premier chapitre passe en revue les différentes technologies et outils utilisés pour la réalisation du projet, et justifie le choix de ces outils vis à vis des autres choix disponibles et concurrents.

Le deuxième et dernier chapitre présente les prototypages de l'application.

Chapitre I : Outils et technologies



Ce chapitre décrit les différents outils et technologies exploités pour la réalisation du projet.



Éléments mis en jeu

- Java / JEE6
- JSF2.0
- EclipseLink, JPA2.0
- EJB3
- PrimeFaces 3.2
- Netbeans 7.1
- Derby
- UML2,PowerAMC 15.

Java EE 6 :

Le but principal de cette version est de poursuivre la simplification de la programmation introduite par Java EE 5. Avec Java EE 5, les EJB, les entités persistantes et les services web ont été remodelés afin d' utiliser une approche plus orientée objet (via des classes Java implémentant des interfaces Java) et pour se servir des annotations pour définir les métadonnées (les descripteurs de déploiement XML sont donc devenus facultatifs). Java EE 6 poursuit dans cette voie et applique les mêmes paradigmes à la couche web. Aujourd' hui, un bean géré par JSF est une classe Java annotée avec un descripteur XML. Java EE 6 s' applique également à simplifier la plate-forme à l' aide de profils et en supprimant certaines technologies obsolètes. Il ajoute des fonctionnalités supplémentaires aux spécifications existantes (en standardisant, par exemple, les beans de session singletons) et en ajoute de nouvelles (comme JAX-RS). Plus que jamais,

les applications Java EE 6 sont portables entre les conteneurs grâce aux noms JNDI standard et à un conteneur EJB intégré.

JSF 2.0 :

JSF 2.0 apporte la simplicité et la richesse aux interfaces utilisateurs dynamiques. Il reconnaît nativement Ajax en fournissant une bibliothèque JavaScript permettant d'effectuer des appels asynchrones vers le serveur et de rafraîchir une page par parties. La création d'interfaces utilisateurs, le contrôle de la navigation dans l'application et les appels synchrones ou asynchrones de la logique métier sont possibles parce que JSF utilise le modèle de conception MVC (Modèle-Vue-Contrôleur). Chaque partie est donc isolée des autres, ce qui permet de modifier l'interface utilisateur sans conséquence sur la logique métier et vice versa.

Java Server Faces est un framework de développement d'applications Web en Java permettant de respecter le modèle d'architecture MVC et basé sur des composants côté présentation

Java Server Faces permet :

- Une séparation de la couche présentation des autres couches (MVC)
- Un mapping entre l'HTML et l'objet
- Un ensemble de composants riches et réutilisables
- Une liaison simple entre les actions côté client de l'utilisateur (eventlistener) et le code Java côté serveur
- Création de nouveaux composants graphiques
- JSF peut être utilisé pour générer autre chose que du HTML (XUL, XML, WML, ...)
- Ajax sans programmation

Primefaces 3.2 :

un jeu de composants puissant et performant open-source supportant Ajax, JSF 2, Push

JPA 2.0 :

JPA 2.0 est un modèle de persistance dont elle conserve l'approche orientée objet utilisant les annotations et, éventuellement, des fichiers de correspondance en XML. Cette seconde version ajoute de nouvelles API, étend JPQL qui est un langage de requête très puissant et indépendant des SGBDR et intègre de nouvelles fonctionnalités.

JPQL :

JPQL (Java Persistence Query Language) sert à définir des recherches d'entités persistantes indépendamment de la base de données sous-jacente. C'est un langage de requête qui s'inspire de SQL (Structured Query Language), le langage standard pour interroger les bases de données relationnelles. La différence principale est que le résultat d'une requête SQL est un ensemble de lignes et de colonnes (une table) Alors que celui d'une requête JPQL est une entité ou une collection d'entités. Sa Syntaxe est orientée objet et est donc plus familière aux développeurs ne connaissant que ce type de programmation. Ils peuvent ainsi manipuler un modèle objet en utilisant la notation pointée classique (maClasse.monAttribut, par exemple) et oublier la structure des tables.

EJB 3.1 :

Une entité EJB3 est un POJO ordinaire dont le mapping est défini par les annotations du JDK5 ainsi que par les annotations EJB3 et/ou tout framework JEE. Il exécute les processus métier ou s'occupe de la persistance des données.

NetBeans 7.1



NetBeans

NetBeans IDE 7.1 est le premier IDE supportant les tout derniers standards et spécifications de la plate-forme Java, avec notamment le support complet du développement pour JavaFX 2.0 et JDK7. Cette nouvelle version offre également le support d'Oracle WebLogic Server 12c, nouvelle édition récemment annoncée par Oracle de son serveur applicatif, conçue pour offrir le support de Java Platform Enterprise Edition (Java EE) 6 aux développeurs qui réalisent des applications sur la plate-forme Oracle WebLogic Server. NetBeans IDE 7.1 intègre également des améliorations importantes pour Swing GUI Builder, le support de CSS3 ainsi que des outils pour le débogage visuel des interfaces utilisateur JavaFX et Swing.

Derby 10.5 :

Initialement nommé Cloudscape, le système de base de données Derby écrit en Java a été offert par IBM à la fondation Apache et est devenu open-source. De son côté, Sun Microsystems a produit sa propre distribution sous le nom de Java DB. Malgré une empreinte mémoire réduite (2 Mo), Derby est un système de base de données relationnelle entièrement fonctionnel qui supporte les transactions et peut aisément s'intégrer dans n'importe quelle solution Java.



Derby a deux modes différents : intégré ou serveur réseau. Le mode intégré correspond au lancement de Derby par une simple application Java mono-utilisateur : en ce cas, il s'exécute dans la même JVM que l'application. C'est ce mode que j'ai utilisé dans mon application car Derby est déjà intégré avec NetBeans. Le 2ème mode c'est le mode serveur réseau correspond au lancement de Derby sous forme de processus séparé, fournissant une connectivité multi-utilisateurs. Nous utiliserons ce mode lorsque nous exécuterons les applications.

GlassFish 3

GlassFish 3, le premier serveur d'application prenant en charge la plate-forme Java Enterprise Edition 6 (Java EE 6) standard. Sa flexibilité améliorée et sa facilité d'utilisation ont permis de réduire les coûts en favorisant la productivité des développeurs, en simplifiant l'architecture d'application et en offrant une prise en

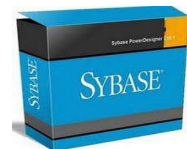
charge dynamique des mises à niveau. Il repose sur une exécution modulaire et flexible basée sur la norme OSGi.

Il permet aux entreprises de créer et de déployer des applications Web à l'aide du profil Web Java EE 6 léger et de facilement exploiter la puissance de la plate-forme Java EE 6 complète pour les applications d'entreprise. Les développeurs bénéficient également du modèle de programmation simplifié et des améliorations de productivité proposés par Java EE 6. Le résultat est une plate-forme flexible apte à offrir une solution ciblée à tout problème, réduisant ainsi le coût et la complexité. GlassFish 3.0.1 utilisant une architecture de micro-noyau basée sur OSGi, les développeurs peuvent commencer avec le profil Web Java EE 6 et utiliser le centre de mise à jour pour effectuer une mise à niveau dynamique vers la plate-forme Java EE 6.

GlassFish 3.0.1 propose également un alignement avec NetBeans 6.9, la prise en charge d'Eclipse, la prise en charge des scripts JRuby/Ruby et Groovy/Graal, une console d'administration conviviale et la connectivité au centre de mise à jour. Une fois que vous aurez installé et commencé à utiliser GlassFish 3.0.1, n'hésitez pas à nous faire part de vos impressions et commentaires.

Power AMC :

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner.



PowerAMC permet de réaliser tous les types de modèles informatiques. Il reste un des seuls qui permet de travailler avec la méthode Merise. cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications.

Chapitre II : Prototypages des interfaces



Ce chapitre présent l'application réalisée durant le stage en présentant un ensemble des interfaces utilisateur.



Commençant par le menu principal de l'administrateur :



Figure 58: menu principale

Le premier élément sur menu c'est Rôle :



Dans Role l'administrateur peut choisir dans le sous menu soit « Ajout » ,dans ce cas la vue qui sera affichée est bien la suivante :

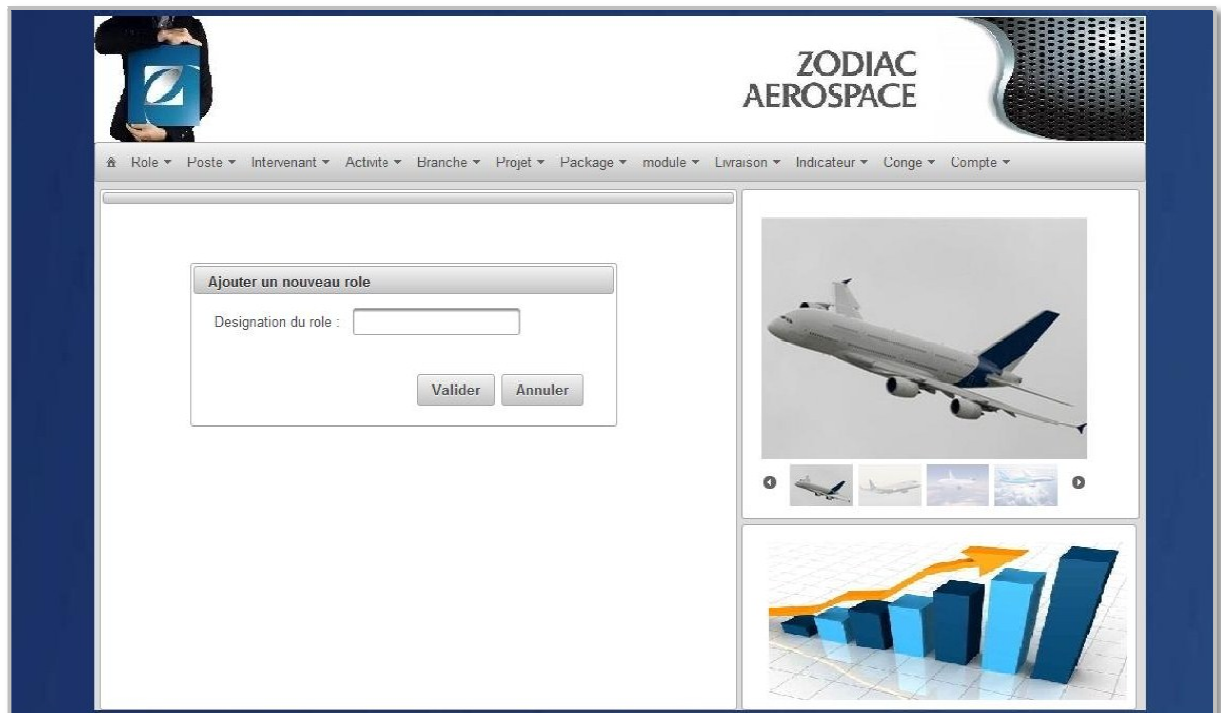


Figure 59: la vue ajout rôle

Dans le cas où l'administrateur saisie un champs déjà existant dans la base de données ,il va avoir le message suivant :

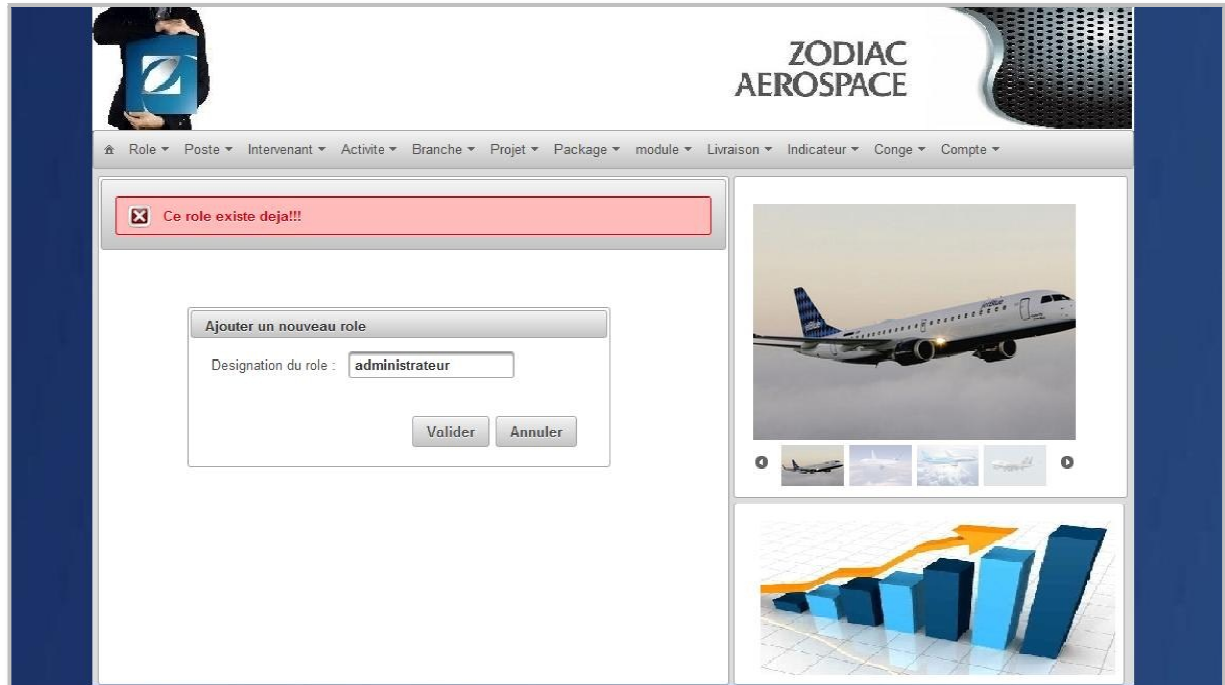


Figure 60: la vue ajout rôle _ validation des champs de saisie

Dans le cas où le champs soit nul , l'administrateur va avoir la vue suivante :

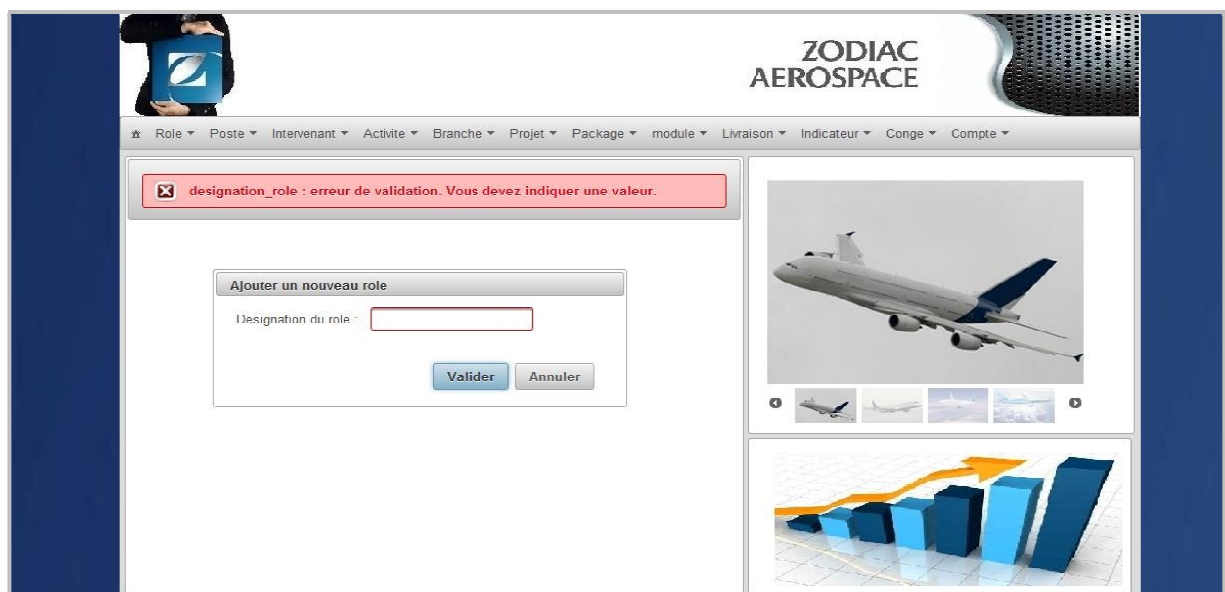


Figure 61: la vue ajout rôle _ validation des champs de saisie2

Dans le cas où le champs saisi par l'administrateur n'est ni nul , ni déjà existant ,la vue de consultation des rôles s'affiche :



Figure 62: la vue consultation des rôles

Si l'administrateur veut modifier le rôle ajouté, il n'a que cliquer sur le pinceau à coté et il va obtenir :

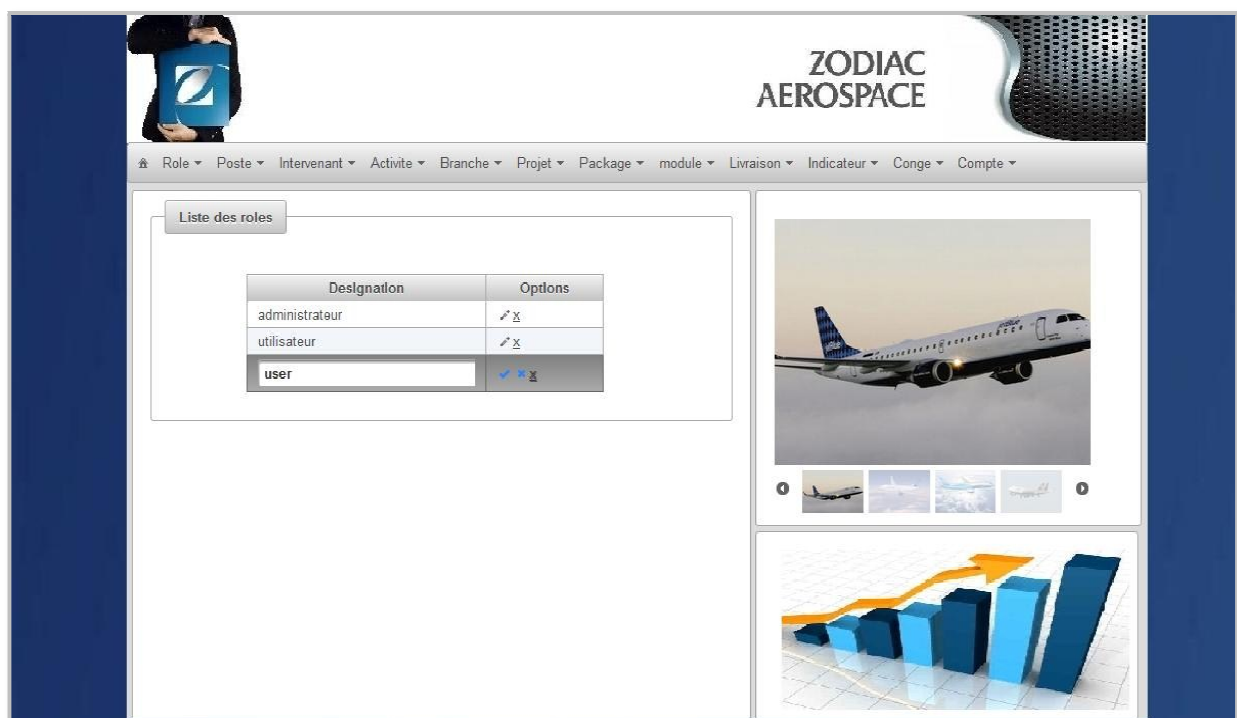


Figure 63: la vue consultation des rôles _ modification d'un rôle

Dans le cas où l'administrateur veut supprimer un rôle ,une boite de confirmation s'affiche :

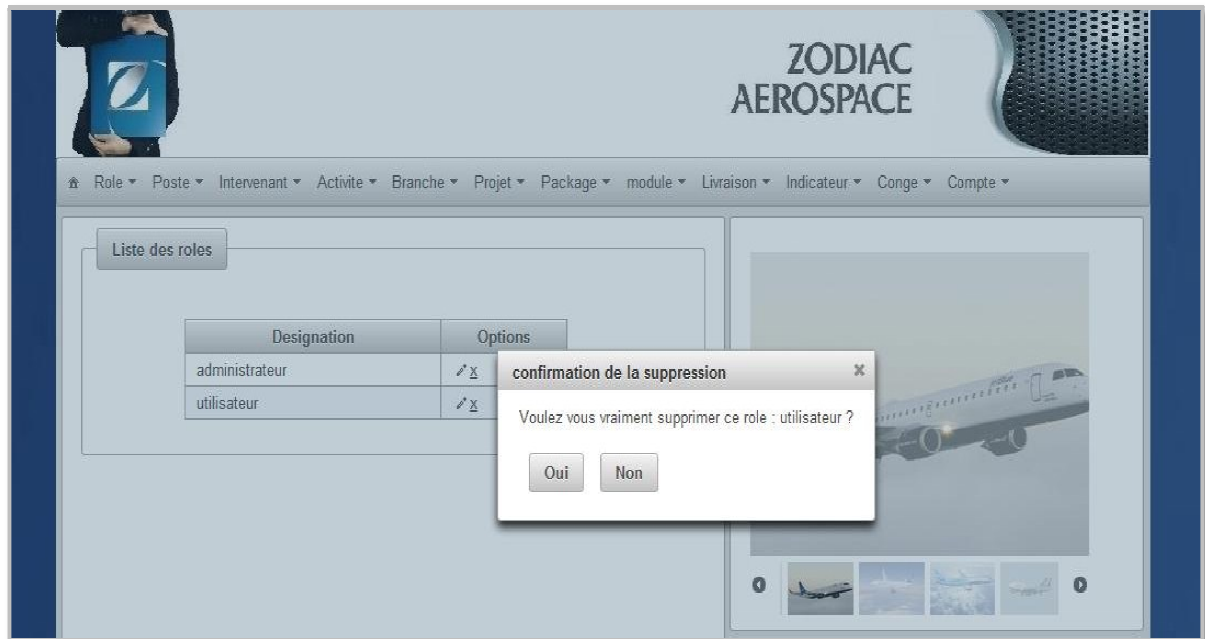


Figure 64: la vue consultation des rôles _ suppression d'un rôle

NB : Ces opérations de modification, de suppression ainsi que les validations sur les champs sont valables pour toutes les autres entités .Alors pour éviter la redondance je ne vais pas les refaire pour entités restantes.

Passant à l'entité Poste :



La vue d'ajout est la suivante :



Figure 65: la vue d'ajout d'un poste

La vue de consultation des postes est la suivante :

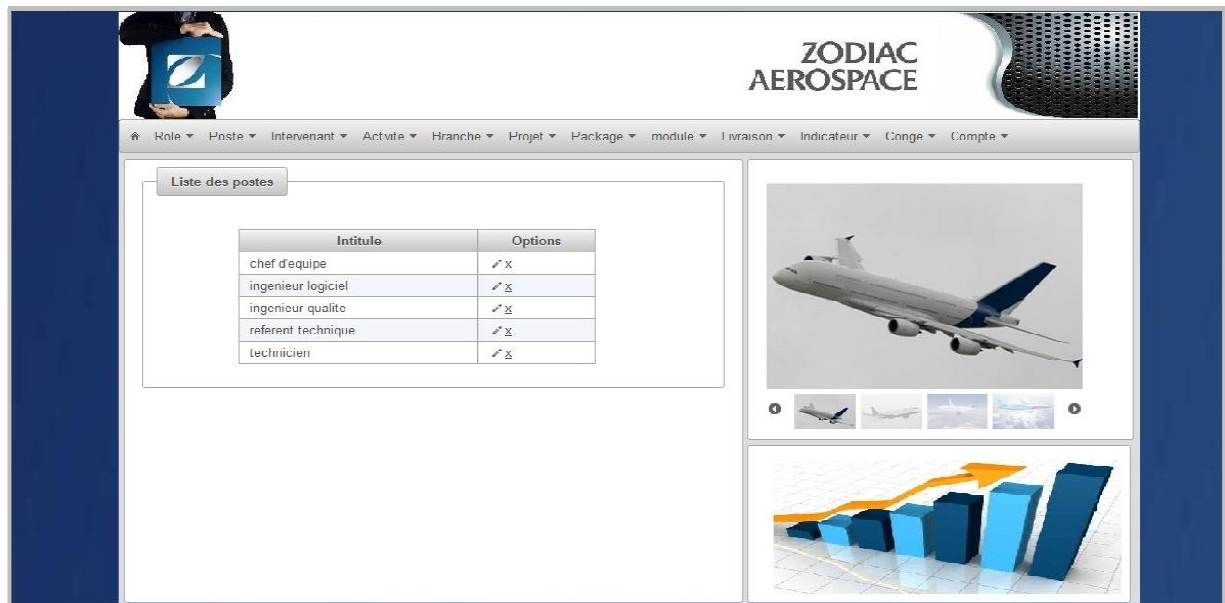
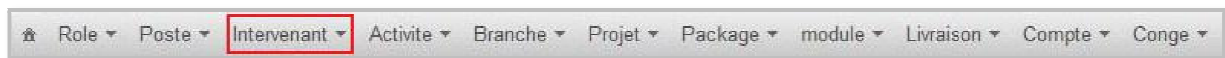


Figure 66: la vue consultation des postes

La 3^{ème} entité est celle des intervenants :



L'ajout d'un intervenant se fait via la vue suivante :

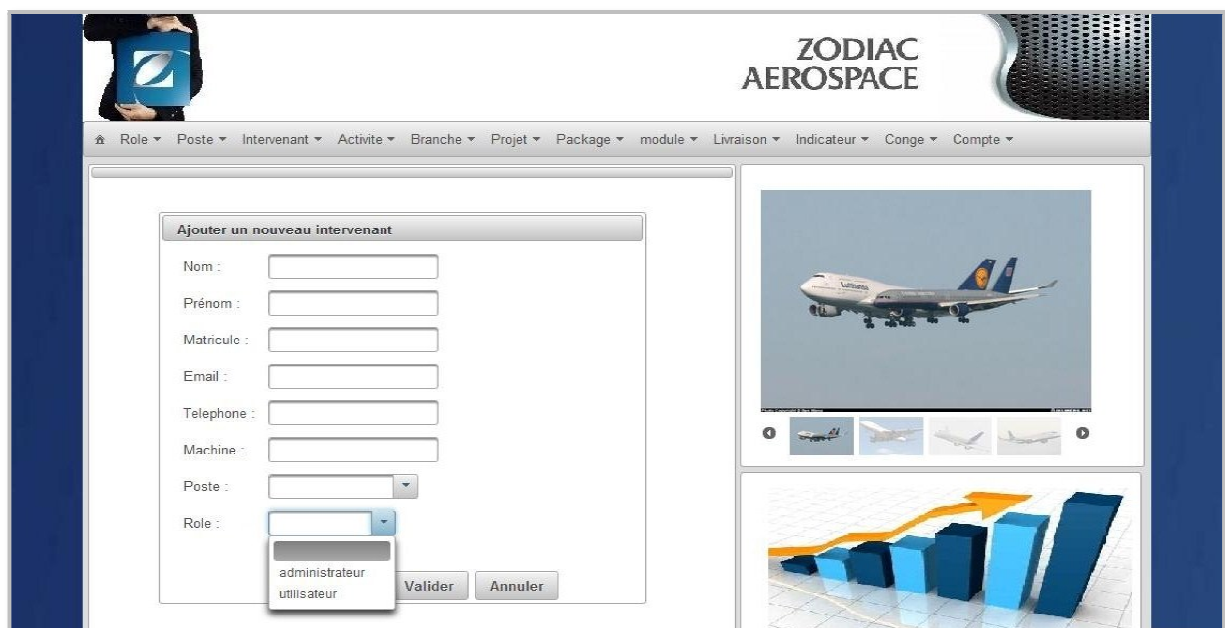


Figure 67: la vue d'ajout d'un intervenant

La consultation se fait via la vue suivante :

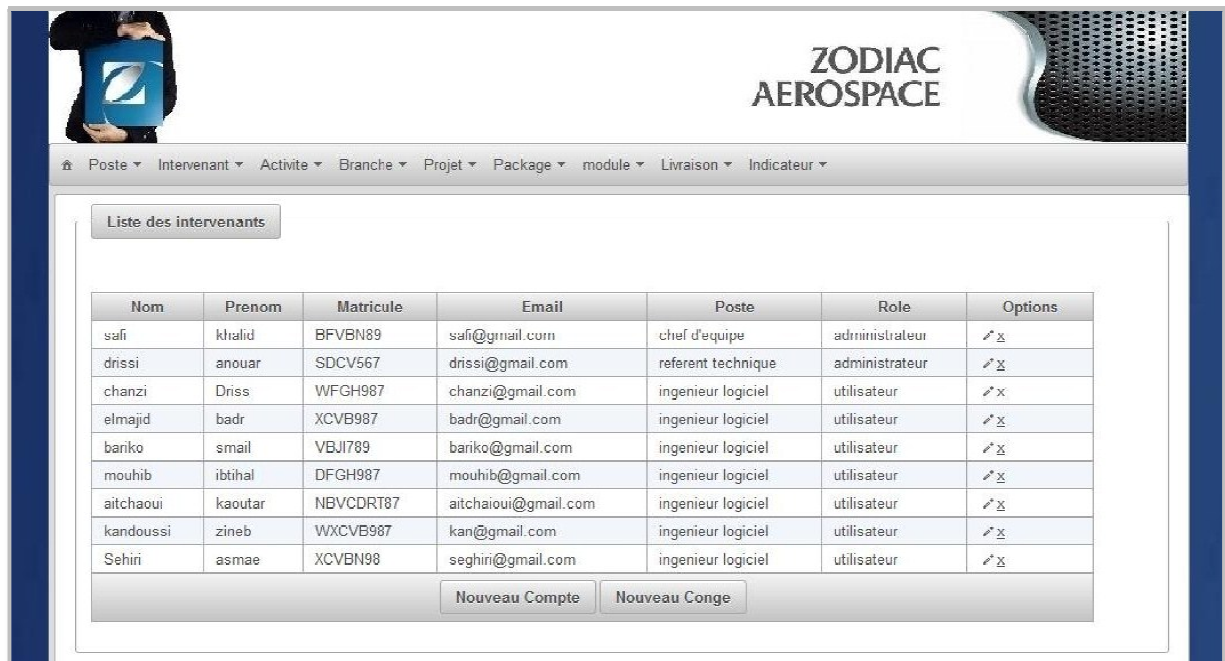


Figure 68: la vue de consultation des intervenants

Cette vue nous permet de consulter, de supprimer ,de modifier un intervenant mais également de créer un nouveau compte et/ou un nouveau congé à chaque'un de ses intervenants .

La création d'un nouveau compte consiste à donner à l'intervenant un login et un mot de passe pour qu'il puisse accéder à l'application ,pour cela :On sélectionne l'intervenant à lequel on veut créer le compte et on clique sur le bouton ' Nouveau Compte' en dessous du tableau comme suit :

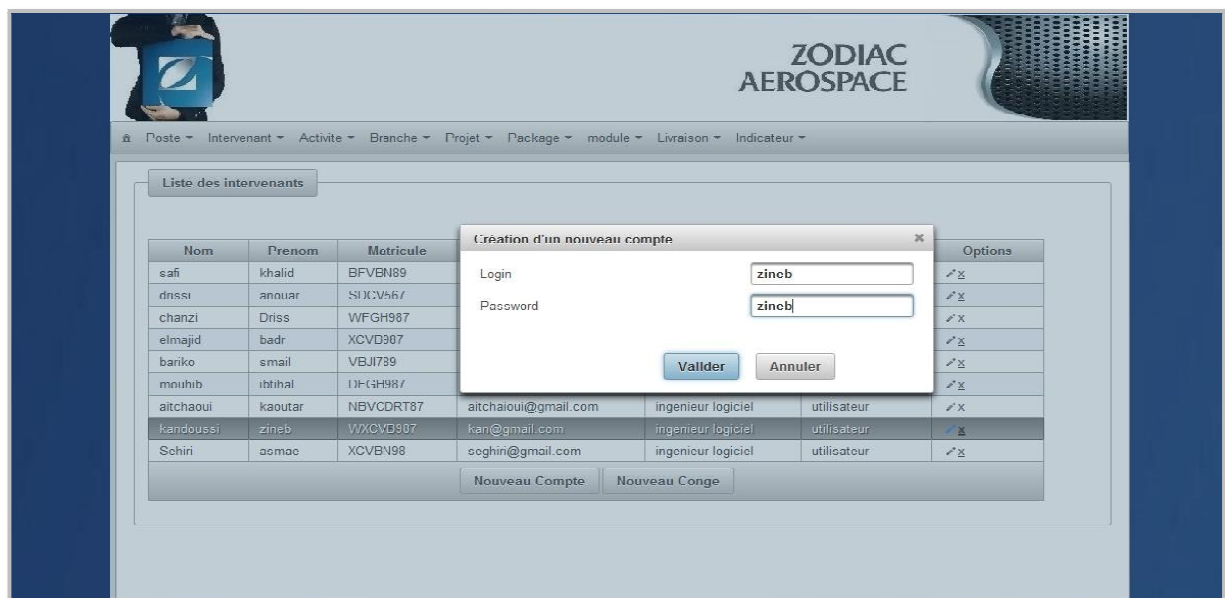
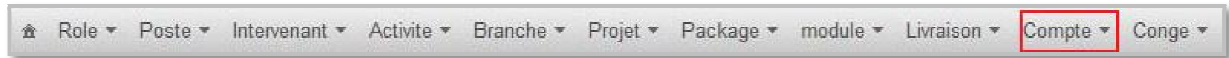


Figure 69: la vue de consultation des intervenants _ création d'un nouveau compte

Si on veut consulter les différents comptes créés ,on clique sur le bouton « consultation Compte » qui est un sous menu de compte :



Et on obtient la fenêtre ci-dessous qui affiche tous les intervenants avec ou sans leurs comptes ,comme ça l'administrateur peut savoir ceux qui n'ont pas encore un compte ,ces derniers ont les deux champs Login et mot de passe vides .

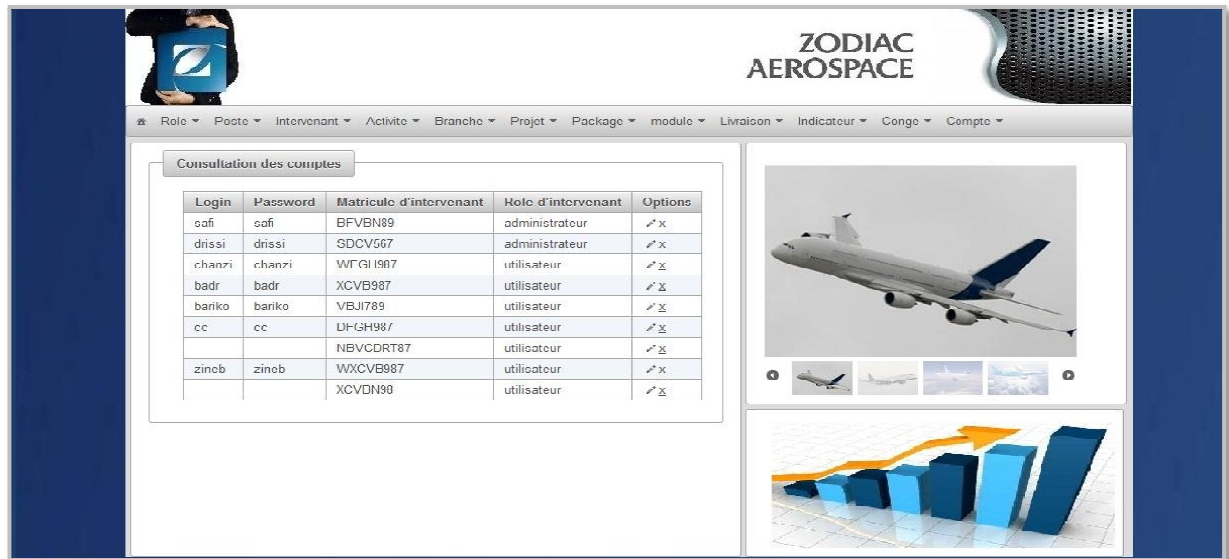


Figure 70: la vue de consultation des comptes

Revenons à la vue de consultation des intervenants. Dans cette vue ,on peut aussi créer un nouveau congé pour les intervenants concernés. Alors si on clique sur le bouton 'Nouveau congé 'en dessous ,on obtient :

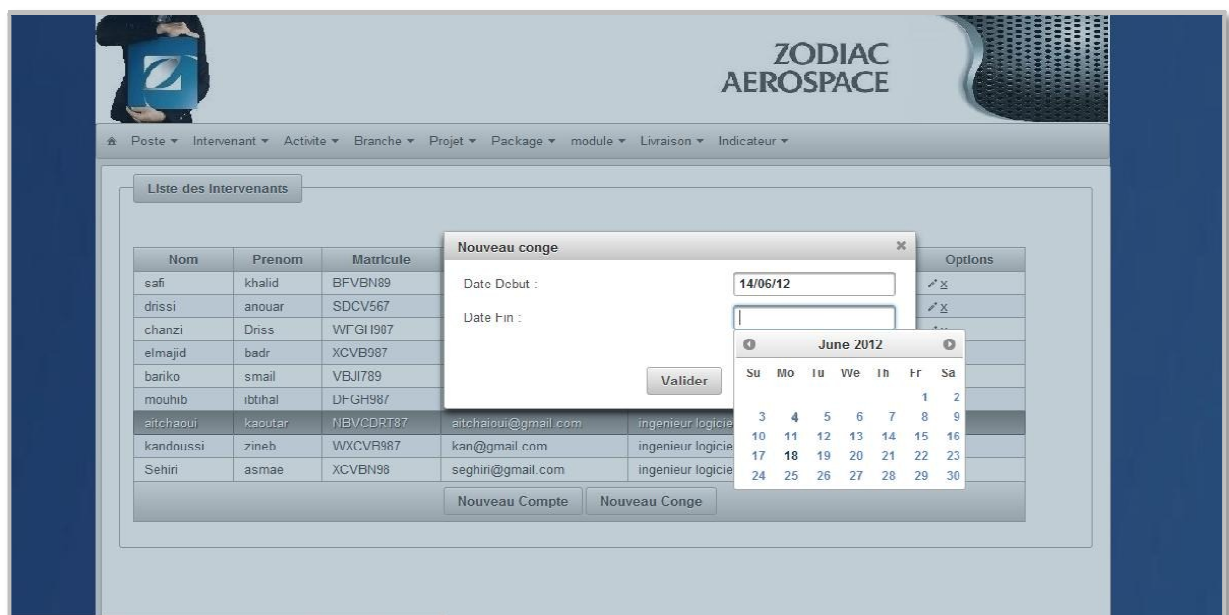
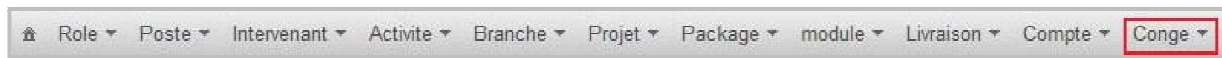


Figure 71: la vue de consultation des intervenants _ création d'un nouveau congé

Si on veut consulter les différents congés, on clique sur l'élément congé du menu .



On clique sur consultation et on obtient :

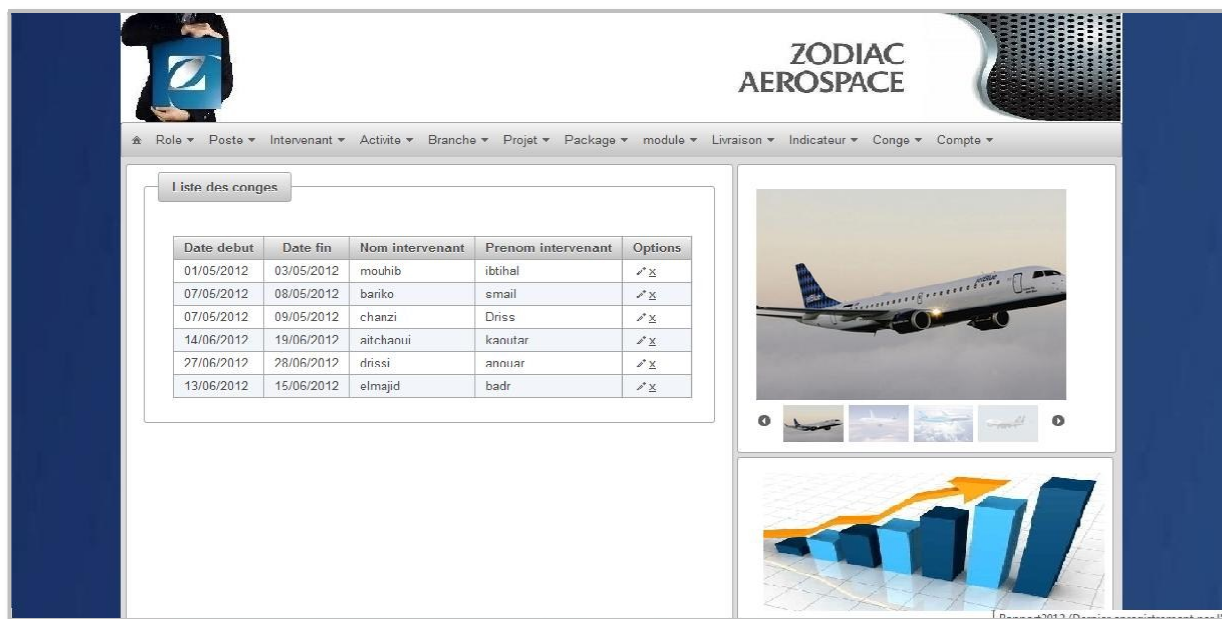


Figure 72: la vue de consultation congés

On passe à l'élément « Activité »:



La vue d'ajout d'une nouvelle activité est la suivante :



Figure 73: la vue d'ajout d'une nouvelle activit

Et celle d'ajout d'un type d'activité est la suivante :

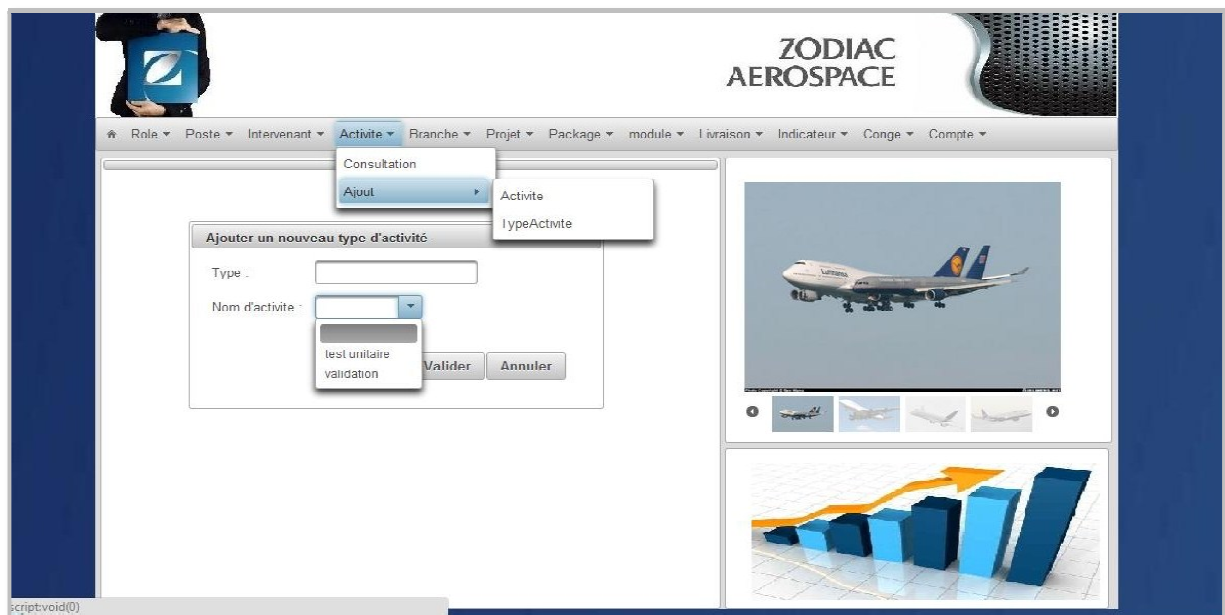


Figure 74: la vue d'ajout d'un nouveau type d'activité

Pour la consultation des activités, on a :



Figure 75: la vue de consultation des activités

On passe à l'entité Branche



La vue d'ajout d'une nouvelle branche est la suivante :



Figure 76: la vue d'ajout d'une nouvelle branche

La vue de consultation est :

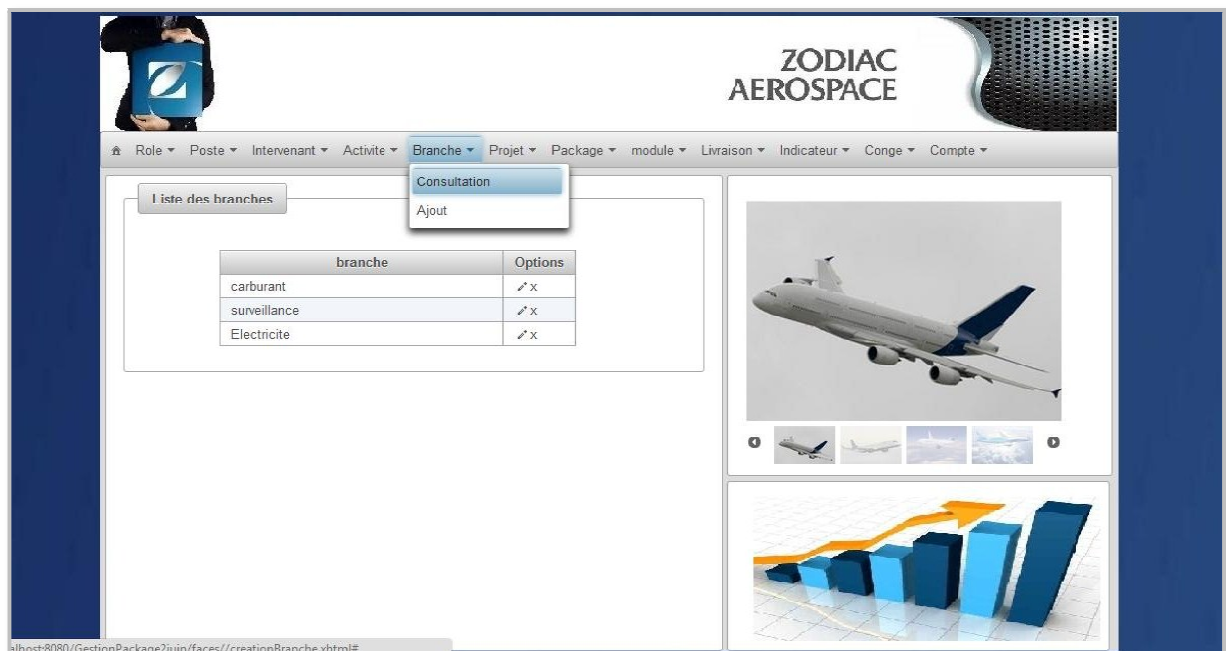


Figure 77: la vue d'ajout d'une nouvelle branche

Ensuite on a l'entité Projet :



Avec la vue d'ajout :

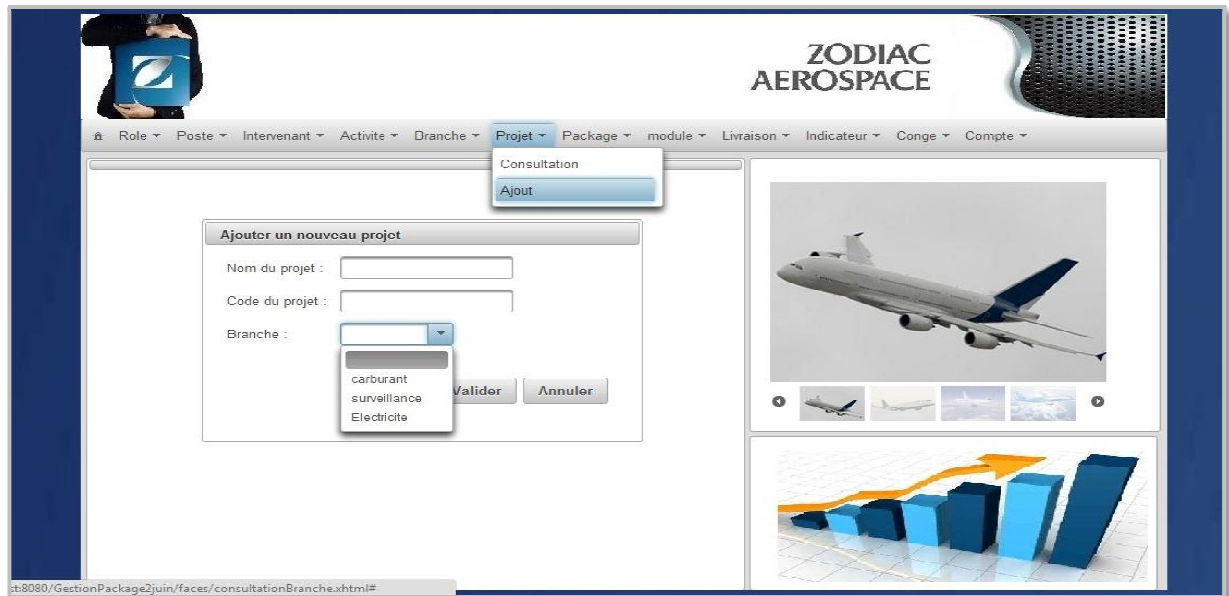


Figure 78: la vue d'ajout d'un nouveau projet

Et la vue de consultation est la suivante :

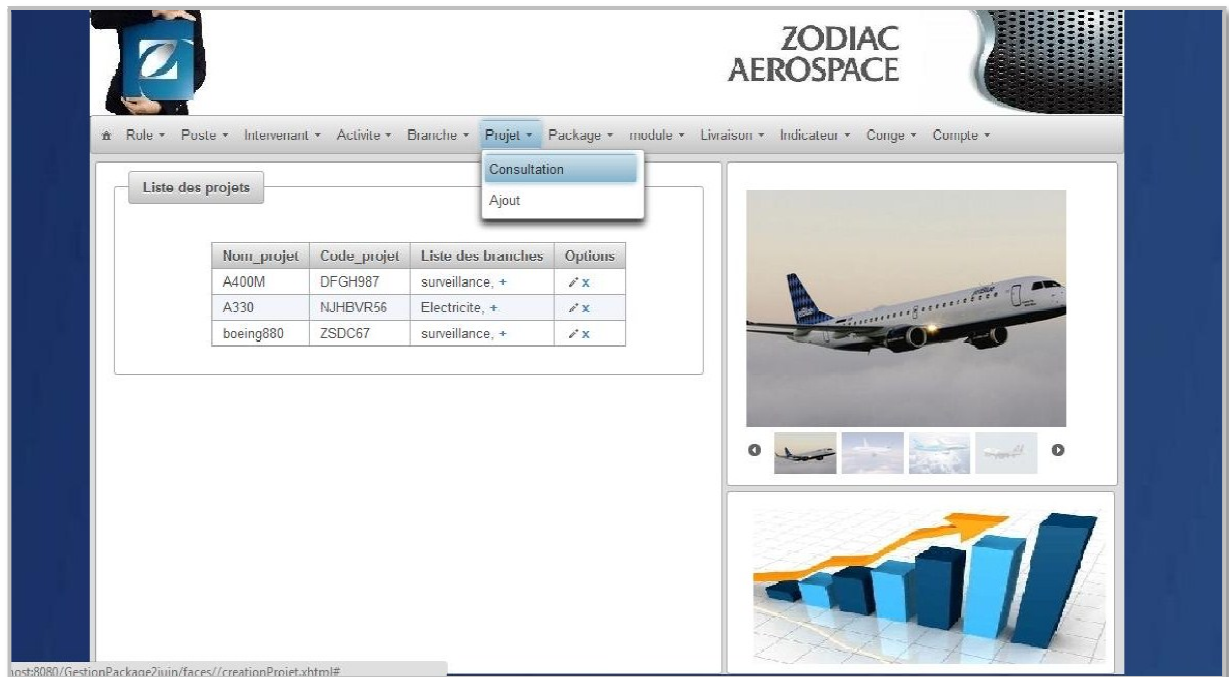


Figure 79: la vue de consultation des projets

Si on veut ajouter une branche à la liste des branches du projet, on clique sur le « + » à côté de la branche, la fenêtre qui s'affiche est la suivante :

On valide et on obtient , la branche carburant qui s'est ajoutée à la branche surveillance pour le projet boeing 880 comme montre la figue ci-dessous :

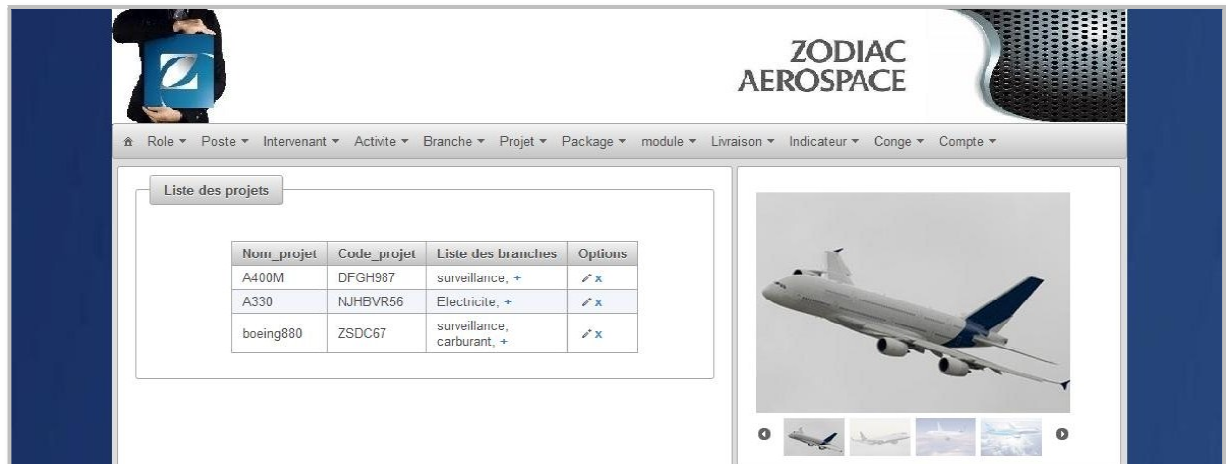
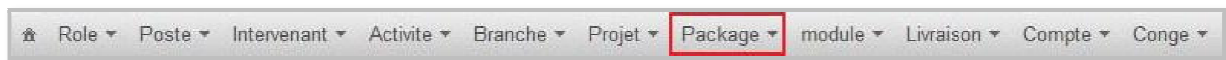


Figure 80: la vue de consultation des projets après l'ajout d'une nouvelle branches à la liste des branches

On passe à l'entité principale « Package » :



Dans cette entité, on peut également ajouter un package via la vue suivante :

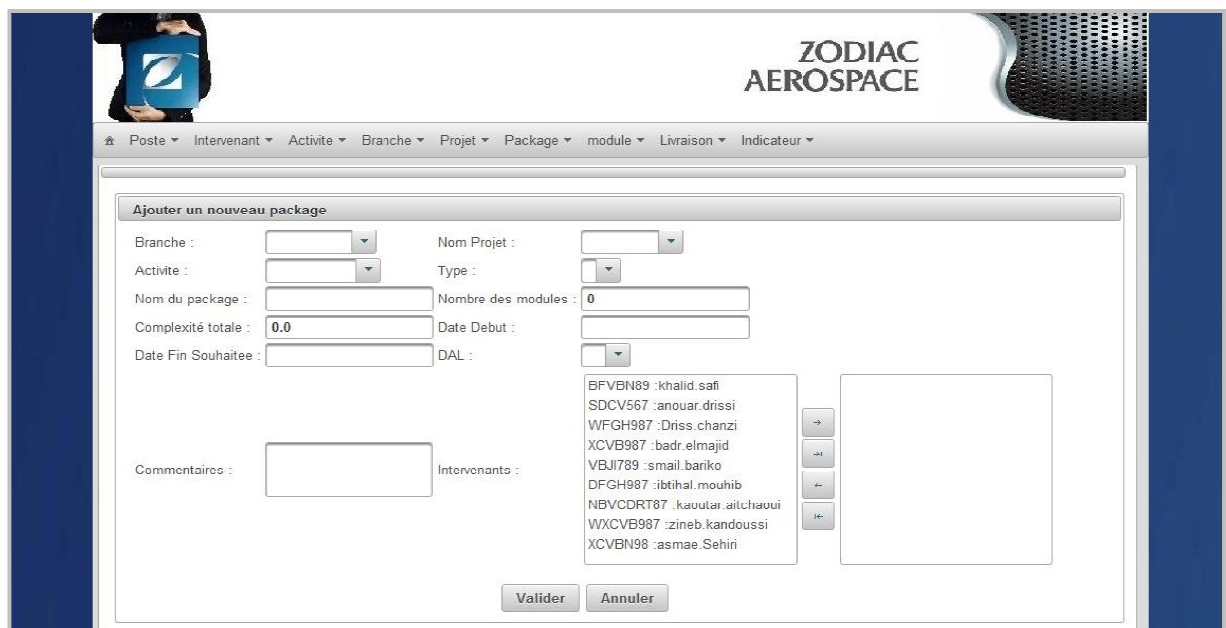


Figure 81: la vue d'ajout d'un nouveau package

Consulter les packages via :

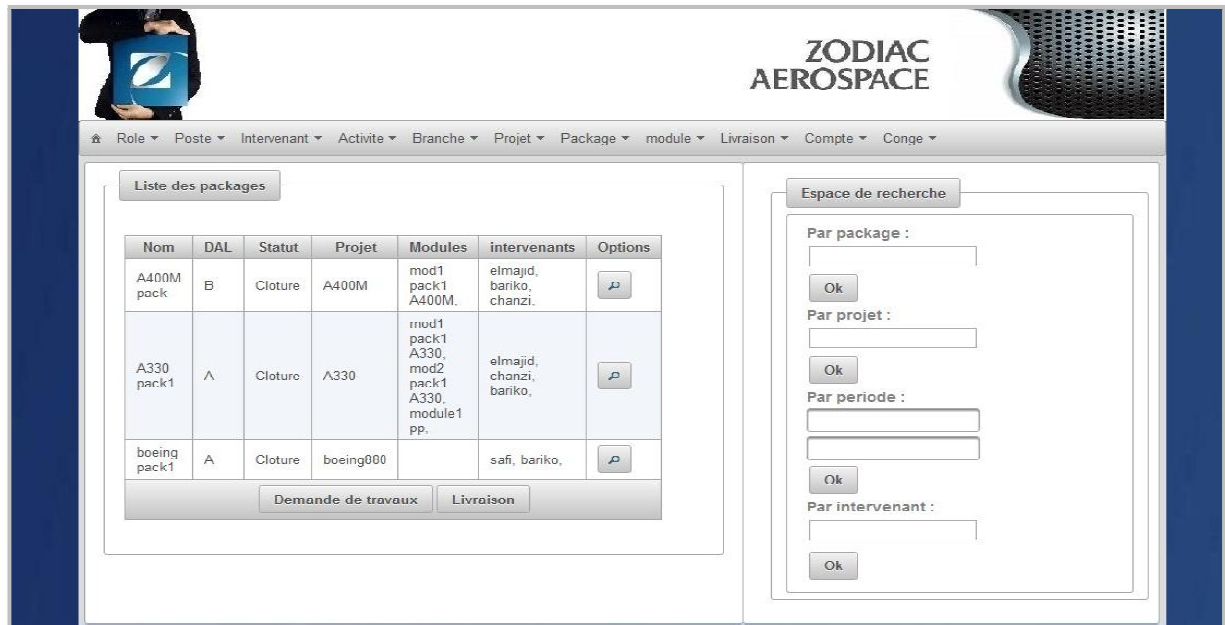


Figure 82: la vue de consultation des packages

Si on veut visualiser toutes les informations du packages on clique sur le bouton à droite du tableau comme suit :



Figure 83: la vue de consultation des packages_visualisation des informations d'un package