

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab. Blida
USDB.

Faculté des sciences.
Département informatique.

**Mémoire pour l'obtention
du diplôme d'ingénieur d'état en informatique.**
Option : Système d'information

Sujet :

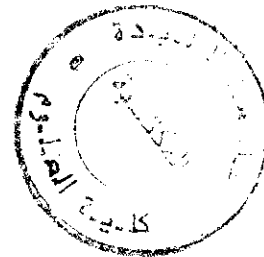
Conception d'un système d'information pour
la commercialisation et gestion du sinistre du
produit « **Assurance Groupe** », et la mise
en place de la solution avec l'architecture 3

Organisme d'accueil :

Algérienne des assurance 2a.

Présenté par : Dechir Salim
Meftouh Kamel

Promoteur : Mme Bensettiti Souad
Encadreur : Mr Meftahi Toufik



- 2005/2006-

MIG-004-124-1

Remerciements

Avant tout, nous remercions Dieu tout puissant de nous donné le morale et la santé pour mener ce modeste travail.

Il est devoir de remercier les enseignants que nous avons eu tout au long de nos années d'étude à l'université de Blida.

Nous adressons nos remerciements à Mme Bensettiti Souad notre promotrice qui nous a été d'une aide précieuse.

Nous tenons à exprimer nos reconnaissance et nos remerciements à Mr Meftahi Toufik , notre encadreur qui a suivi et guidé avec attention et bienveillance l'élaboration de ce travail.

Aussi, nous tenons à remercier vivement les responsables de la direction système d'information au niveau de l'algérienne des assurance 2a :le directeur Douadji ahmed, Achir, Rachid, yazid, Ahmed, Amrane et Hichème sans oublier monsieur Koukhi responsable de l'assurance groupe au niveau de la 2a et Mme souad :responsable de l'assurance groupe au niveau de l'agence 1622.

Merci

Dédicace

Je dédie ce modeste travail à :

✚ Mes chers parents qui m'ont donné l'amour, la force et tout le courage
qu'il faut pour mener a terme ce travail et surtout pendant mon long chemin
Pour que je sois ce que je suis aujourd'hui.

✚ A Ma grande famille et surtout ma grande mère

✚ A mes chers frères et sœur

✚ A tous mes amies

✚ A mes amis de l'université de Blida qui ont partagé avec moi les meilleurs moments.

✚ A mon binôme et sa famille ;

Kamel

Dédicace

A mes très chers parents ;

Aucun acte ne nous pourra exprimer mes sentiments envers vous.

A mes chers frères et sœurs ;

Qui m'encouragent toujours.

A mes grands parents ;

A toute la famille ;

A mes amis de l'université de Blida qui ont partagé avec moi les meilleurs moments.

A mon binôme et sa famille ;

A tous ...

Salim

TABLE DE MATIERES

INTRODUCTION.....	1
PROBLEMATIQUE.....	2
Chapitre I : Organisme d'accueil	
I. PRESENTATION DE L'ORGANISME D'ACCEUIL.....	4
I.1 Historique.....	4
I.2 Mission et objectif.....	4
I.3 Organigramme de l'algériennes des assurances 2a.....	5
I.4 Organisation des délégations de l'algériennes des assurances 2a	5
II PRESENTATION DE LA STRUCTURE D'ACCEUIL.....	7
II.1 Mission et objectif.....	7
II.2 Organisation des agences.....	7
Chapitre II : Etude de l'existant	
I. INTRODUCTION.....	9
II. LEXIQUE.....	9
III.	
II.1 Définition	9
II.2 Autres définitions.....	9
IV. PRESENTATION DE LA STUATION ACTUEL.....	14
III.1 Le flux d'information.....	14
III.2 Etude des postes de travail, des documents, des registres.....	17
III.2.1 Etude des postes de travail.....	17
III.2.2 Etude des documents.....	20
III.2.3 Etude des registres.....	32
III.3 Etude de codification.....	33
III.3.1 Définition.....	33
III.3.2 Règles.....	33
III.3.3 Codification existantes.....	33
IV DIAGNOSTIQUE.....	35
Chapitre III : Conception	
I. INTRODUCTION.....	37
II. PRESENTATION DE L'UML.....	37
II.1 Introduction.....	37
II.2. La modélisation.....	37
II.3. Apport de la modélisation objet.....	37
II.4. Définition de l'UML.....	40
II.5. Les objectifs d'UML.....	40
II.6. UML un langage.....	40
II.7. UML un langage pour	40
II.8. les domaines d'utilisation d'UML.....	40
II.9. Les éléments de base en UML.....	41
II.10. Les différents diagrammes d'UML.....	41
II.11. Les façons de voir un système	41
II.12. Description des 9 diagrammes	42
1. Diagramme de Classes.....	42

2. Diagramme d'Objets.....	44
3. Diagramme de Collaboration	45
4. Diagramme de séquence.....	45
5. Diagramme d'état transitions.....	46
6. Diagrammes d'activités.....	46
7. Le modèle des cas d'utilisation.....	47
8. Diagramme de composant	48
9. Diagramme de déploiement	48
III. La CONCEPTION DU NOUVEAU SYSTEME EN UTILISANT UML	50
III.1. modèle fonctionnel.....	50
1. les cas d'utilisations.....	50
1.1 les acteurs.....	50
1.2. La liste des cas d'utilisation.....	51
1.3 Les diagrammes des cas d'utilisations	52
2. les diagrammes des séquences.....	61
3. les diagrammes d'activités.....	69
4. les diagrammes de collaboration.....	76
III.2 modèle logique.....	80
1. définition des classes.....	80
1.1 Présentation de la nouvelle codification.....	82
1.2 Le diagramme de classe.....	84
III.3. le modèle physique.....	85
1. le diagramme de déploiement.....	85

Chapitre IV : Présentation de l'architecture 3 tiers

I. INTRODUCTION.....	87
II LES TROIS NIVEAUX D'ABSTRACTION D'UNE APPLICATION	87
III. L'ARCHITECTURE UN TIER.....	88
III.1. Présentation.....	88
III.2.Limitations l'architecture un tiers	90
IV. L'ARCHITECTURE DEUX TIERS.....	90
IV.1. Présentation	90
IV.2 Le dialogue client-serveur	90
IV.3.Limites du client-serveur deux tiers.....	93
V. L'ARCHITECTURE TROIS TIERS.....	93
V.1.Système d'information et l'architecture trois-tiers	93
V.2. Objectifs.....	94
V.3.caractéristiques de l'architecture 3-tiers	94
V.4 Principe de l'architecture trois-tiers	95
V.5. Les bases de données et le web	95
V.5.1 .La révolution Internet	95
V.5.1. 1.Les standard d'Internet	95
V.5.1.2. Adaptation à l'entreprise : Intranet.....	97
V.5.2. Répartition des traitements	97
V.5.3. Le client léger	99

V.5.3.1. Présentation	99
V.5.3.2. Ergonomie	99
V.5.3.3. Avantages de l'architecture 3-tiers	101
V.5.3.4. Conclusion	102

Chapitre V : Implémentation

I. LES OUTILS DE DEVELOPPEMENT	104
II. PRESENTATION DE LA PLATE FORME ORACLE	104
II.1 le serveur de données	104
II.2 le serveur d'application Oracle 9iAS	106
II.2.1 Oracle 9i Forms developer	106
II.2.2 Oracle 9iAS Forms services	107
II.2.3 Aperçu du processus d'exécution d'une application Oracle 9iForms	108
II.2.4. Oracle Jinitiator	110
II.3 le poste client	110
III. REALISATION ET MISE EN OUVRE DE L'APPLICATION	111
III.1 Etapes de réalisations	111
III.2 Présentation de l'application	112
IV. CONCLUSION	127
CONCLUSION GENERALE	128

TABLEAUX

Tableau II.1: Description du graphe de flux d'information.....	16
Tableau II.2: Description de la codifications des documents et registres.....	17
Tableau II.3: Description de la fiche de fonction N=°1.....	18
Tableau II.4: Description de la fiche de fonction N=°2.....	19
Tableau II.5: Description de la fiche de document N=°1.....	20
Tableau II.6: Description de la fiche de document N=°2.....	21
Tableau II.7: Description de la fiche de document N=°3.....	22
Tableau II.8: Description de la fiche de document N=°4.....	23
Tableau II.9: Description de la fiche de document N=°5.....	24
Tableau II.10: Description de la fiche de document N=°6.....	25
Tableau II.11: Description de la fiche de document N=°7.....	26
Tableau II.12: Description de la fiche de document N=°8.....	27
Tableau II.13: Description de la fiche de document N=°9.....	28
Tableau II.14: Description de la fiche de document N=°10.....	29
Tableau II.15: Description de la fiche de document N=°11.....	30
Tableau II.16: Description de la fiche de document N=°12.....	31
Tableau III.1 : listes des cas d'utilisation	51
Tableau III.2 : définition des classes.....	80

FIGURES

Figure I.1 :Organigramme de l'algérienne des assurances2.....	5
Figure I.2 : Organigramme des délégations l'algérienne des assurances2.....	6
Figure I.3 : L'organisation des agences.....	7
Figure II.1: graphe générale de flux d'information.....	7
Figure.III. 1 – Historique de la constitution d'UML.....	38
Figure. III. 2 – les diagrammes UML.....	41
Figure.III.3 – vues de kruchten.....	41
Figure.III.4. – Exemple de diagramme de classe.....	42
Figure. III. 5 – Exemple de calsses association.....	44
Figure .III. 6 – Exemple de diagramme d'objet.....	44
Figure .III. 7 – Exemple de diagramme de collaboration.....	45
Figure.III. 8 – Exemple de diagramme de séquence.....	45
Figure.III. 9 – Exemple de diagramme d'etat-transition.....	46
Figure.III. 10 – Exemple de diagramme d'activité.....	46
Figure.III. 11 – Exemple de diagramme de cas d'utilisation.....	47
Figure.III. 12 – Exemple d'extension et d'utilisation.....	47
Figure.III.13. Exemple diagramme de composants.....	48
Figure.III.14. Exemple diagramme de déploiement.....	48
Figure.III.15.diagramme cas d'utilisation « Authentification ».....	47
Figure.III.16.diagramme cas d'utilisation « Configurer mot de passe».....	51
Figure.III.17.diagramme cas d'utilisation « Créer agence,délégation ».....	52
Figure.III.18.diagramme cas d'utilisation « supprimer agence,délégation ».....	52
Figure.III.19.diagramme cas d'utilisation « Créer police ».....	53
Figure.III.20.diagramme cas d'utilisation « Créer avenant ».....	53
Figure.III.21.diagramme cas d'utilisation « Créer sinistre ».....	54
Figure.III.22.diagramme cas d'utilisation « Créer remboursement ».....	54
Figure.III.23.diagramme cas d'utilisation « consulter police ».....	55
Figure.III.24 diagramme cas d'utilisation « consulter Avenant ».....	55
Figure.III.25.diagramme cas d'utilisation « consulter remboursement ».....	56
Figure.III.26 diagramme cas d'utilisation « consulter Adhérent ».....	56
Figure.III.27 diagramme cas d'utilisation « supprimer police/adhérent ».....	57
Figure.III.28 diagramme cas d'utilisation « consulter agence ».....	57
Figure.III.29 diagramme cas d'utilisation « consulter délégation ».....	58
Figure.III.30 diagramme des cas d'utilisation globale.....	59
Figure.III.31 diagramme de séquence« identification par succès».....	60
Figure.III.32diagramme de séquence« identification avec erreur».....	61
Figure.III.33 diagramme de séquence« ajouter police».....	61
Figure.III.34.diagramme de séquence« ajouter avenant».....	62
Figure.III.35 diagramme de séquence« ajouter sinistre».....	62
Figure.III.36.diagramme de séquence« ajouter remboursement».....	63
Figure.III.37.diagramme de séquence« ajouter agence/délégation».....	64
Figure.III.38.diagramme de séquence« configurer mot de passe».....	64
Figure.III.40 diagramme de séquence« supprimer agence/délégation».....	65
Figure.III.41diagramme de séquence« supprimer police».....	65

Figure.III.42 diagramme de séquence« consulter police».....	66
Figure.III.43diagramme de séquence« ajouter avenant».....	66
Figure.III.44 diagramme de séquence« consulter agence».....	67
Figure.III.45 diagramme de séquence« consulter remboursement».....	67
Figure.III.46 diagramme d' activité« produire police».....	68
Figure.III.47 diagramme de activité« produire avenant».....	69
Figure.III.48 diagramme de activité« déclarer sinistre».....	70
Figure.III.49 diagramme de activité« remboursement».....	71
Figure.III.50 diagramme de activité« créer délégation».....	72
Figure.III.51 diagramme de activité« créer agence».....	73
Figure.III.52 diagramme de activité« consulter».....	74
Figure.III.53 diagramme de collaboration« police».....	75
Figure.III.54 diagramme de collaboration« Avenant».....	75
Figure.III.55 diagramme de collaboration« sinistre».....	76
Figure.III.56 diagramme de collaboration« Remboursement».....	76
Figure.III.57 diagramme de collaboration« police».....	77
Figure.III.58 diagramme de collaboration« agence».....	77
Figure.III.59diagramme de collaboration« délégation».....	78
Figure.III.60 diagramme de classe».....	83
Figure.III.61 diagramme de déploiement».....	86
Figure V.1 Architecture physique du système	104
Figure V.2 Architecture d'Oracle 9i FORMS.....	108
Figure V.3. Authentification.....	112
Figure V.4 fenêtre principale de producteur.....	113
Figure V.5 Police.....	114
Figure V.6 Adhérent.....	115
Figure V.7 Garantie.....	116
Figure V.8 Prime.....	117
Figure V.9 Avenant.....	118
Figure V.10 Sinistre.....	119
Figure V.11 Remboursement.....	119
Figure V.12 Consultation.....	120
Figure V.13 Consultation police.....	121
Figure V.14 Consultation Avenant.....	122
Figure V.15 Consultation Sinistre.....	123
Figure V.16 Consultation Client.....	123
Figure V.17 délégation.....	124
Figure V.18 Consultation délégation.....	124
Figure V.19 Agence.....	125
Figure V.20 Consultation Agence.....	125
Figure V.21 Producteur.....	126
Figure V.21 Configurer mot de passe.....	126

Introduction :

Les entreprises sont actuellement confrontées à une concurrence globale et effrénée. L'ouverture des marchés est une réalité. Cette ouverture offre naturellement de formidables opportunités de croissance, mais elle apporte aussi de nouveaux concurrents.

L'entreprise a besoin d'analyser son activité au jour le jour, de façon proactive et pertinente, afin d'augmenter ses performances. Elle va pour cela définir les indicateurs qui lui conviennent et, en fonction des résultats donnés par ces indicateurs, corriger certaines faiblesses, puis affiner les indicateurs pour aller de plus en plus loin dans l'amélioration de sa productivité.

Une entreprise aujourd'hui ne peut plus se permettre de naviguer à vue, en mode réactif. Elle ne peut plus se permettre d'attendre de rencontrer des écueils pour agir. Elle a constamment besoin d'anticiper et, pour cela, il lui faut des instruments qui sachent travailler en temps réel, et qui vont l'aider à réagir vite et à gagner en compétitivité.

A l'aube de cette ère de l'information, l'algérienne des assurances 2a comme toute entreprise doit se doter d'une architecture pour son système d'information. Les pièges et défis sont nombreux et variés: fiabilité, sécurité et intégration des données et évolutivité des applications. Pour répondre à ces enjeux, il est nécessaire de disposer d'une infrastructure ouverte sur un ensemble de standards qui favorise le plus possible l'approche composants Web. A l'image de ce que l'on trouve dans le développement logiciel, on entend par composants Web des unités de développement et de déploiement présentant des interfaces et des services bien définis. Le développeur d'applications en s'appuyant sur cet éventail de fonctionnalités *normalisées* se concentre sur le développement de l'application elle-même. Cette approche ne laisse pas l'industrie indifférente! C'est la seule, dans la limite des technologies dont nous disposons aujourd'hui, qui soit en mesure de répondre aux problématiques de déploiement, de sécurité, d'administration et de changement de facteur d'échelle. Dans ce cadre il est nécessaire de disposer d'un environnement devant assurer la cohérence de cet ensemble.

Ce projet de fin d'étude a pour objectifs : Conception d'un système d'information pour la commercialisation et gestion du sinistre du produit « **Assurance Groupe** », et la mise en place de la solution avec l'architecture 3 tiers.

Le présent rapport de notre PFE, comporte outre l'introduction, la conclusion et la bibliographie, les quatre chapitres suivants :

- Premier chapitre « Organisme d'accueil » : le contexte dans lequel s'est déroulé le stage en décrivant l'entreprise et son organisation
- Deuxième chapitre « étude de l'existant » : nous avons exposé le système utilisé par l'entreprise.
- Troisième chapitre « conception » : proposition de nouveau système.
- Quatrième chapitre « architecture 3-tier » : présentation de l'architecture 3-tier ainsi ses avantages.
- Cinquième chapitre « implémentation » qui décrit la gamme des outils utilisés pour développer le nouveau système.

La problématique :

Pour l'algérienne des assurances 2a, l'agence représente l'unité de production dans une usine. En effet, les responsables de la 2a font tout leur possible pour perfectionner le fonctionnement de l'agence par le développement des outils de travail en utilisant des méthodes modernes. Le présent sujet sert à trouver des solutions à deux types de problèmes principaux :

1. les problèmes de traitement :

- De façon générale, l'algérienne des assurances 2a n'est pas arrivée à une informatisation totale des outils de travail qu'implique l'existence de problèmes traditionnels.
- L'existence des difficultés de gestion à cause du nombre important de clients qui accroit jour après jour.
- Le mécanisme actuel de traitement a posteriori des demandes de remboursement présentent des limites qui ne permettent pas de garantir dans tous les cas, que seul les personnes ayant droit aux services assurés en bénéficient.

2. Le problème de coût :

- L'algérienne des assurances 2a est une entreprise représentée par des agences dispersées sur le territoire national et il faut noter que son réseau s'étend jour après jour, le problème de déploiement des applications informatiques se pose parmi les obstacles, en plus de la maintenance et de la mise à jours des applications.

Chapitre I: Organisme d'accueil

I. Présentation de l'organisme d'accueil:

I.1. Historique:

La loi de l'ordonnance n° 07/95 du 25 janvier 1995 relative aux assurances traduit la volonté et les efforts initiés par les pouvoirs publics pour relancer l'économie ; ils s'inscrivent, entre autres, dans le mouvement de recomposition actuel du secteur des assurances. La création de l'Algérienne des assurances 2a en est une conséquence directe.

Créée dans le cadre de l'ordonnance n° 07/95 du 25 janvier 1995 consacrant l'ouverture du marché des assurances. L'Algérienne des assurances a obtenu son agrément le 05/08/1998 sous le numéro 14/98.

C'est une société par action au capital social de 1.000.000.000.00 DA entièrement libéré avec un actionariat constitué par un groupe d'investisseurs d'envergure activant dans des secteurs d'activité variés.

En terme de prestation, son intervention vient en complément des services proposés par les entreprises déjà opérationnelles et s'inscrit dans l'élargissement du choix offert aux publics.

I.2. Mission et objectif:

En terme de segment du marché et parallèlement aux risques traditionnels, les cibles privilégiées sur lesquels la 2a concentra ces efforts pour diffuser l'assurance au sein du publics sont les gisement non encor exploités ou très peu développés, à savoir les assurances des particuliers et des ménages, les assurances des personnes à caractère de prévoyances et d'épargne, les assurances des PME/PMI , les produits nouveaux qui accompagne l'évolution de l'économie et de la société(assurance temporaire en cas de décès,risques électronique,assurances monétiques, assurance retrait ,maladie complémentaire et l'assurance caution).

Cernant les moyens mis en œuvres, la 2a s'appuie sur un encadrement constitué de professionnels de haut niveau. Elle procède à la mises à niveau puis à la formation permanente des agents de la société pour assurer un haut degré de professionnalisme et une amélioration continue du service offert, avec une utilisation forte de technique moderne de gestion, notamment l'outil informatique, et une orientation centrée sur le service client.

La 2a a pour objectifs premier d'améliorer les prestations offertes aux clients en terme de:

- Information sur le choix du produit le mieux adapté.
- Conseils pour l'amélioration des risques par une priorité accordée à la prévention.
- Etudes globales des risques.
- Diligence dans le règlement des sinistres.

I.3. Organigramme de l'algérienne des assurances2a:

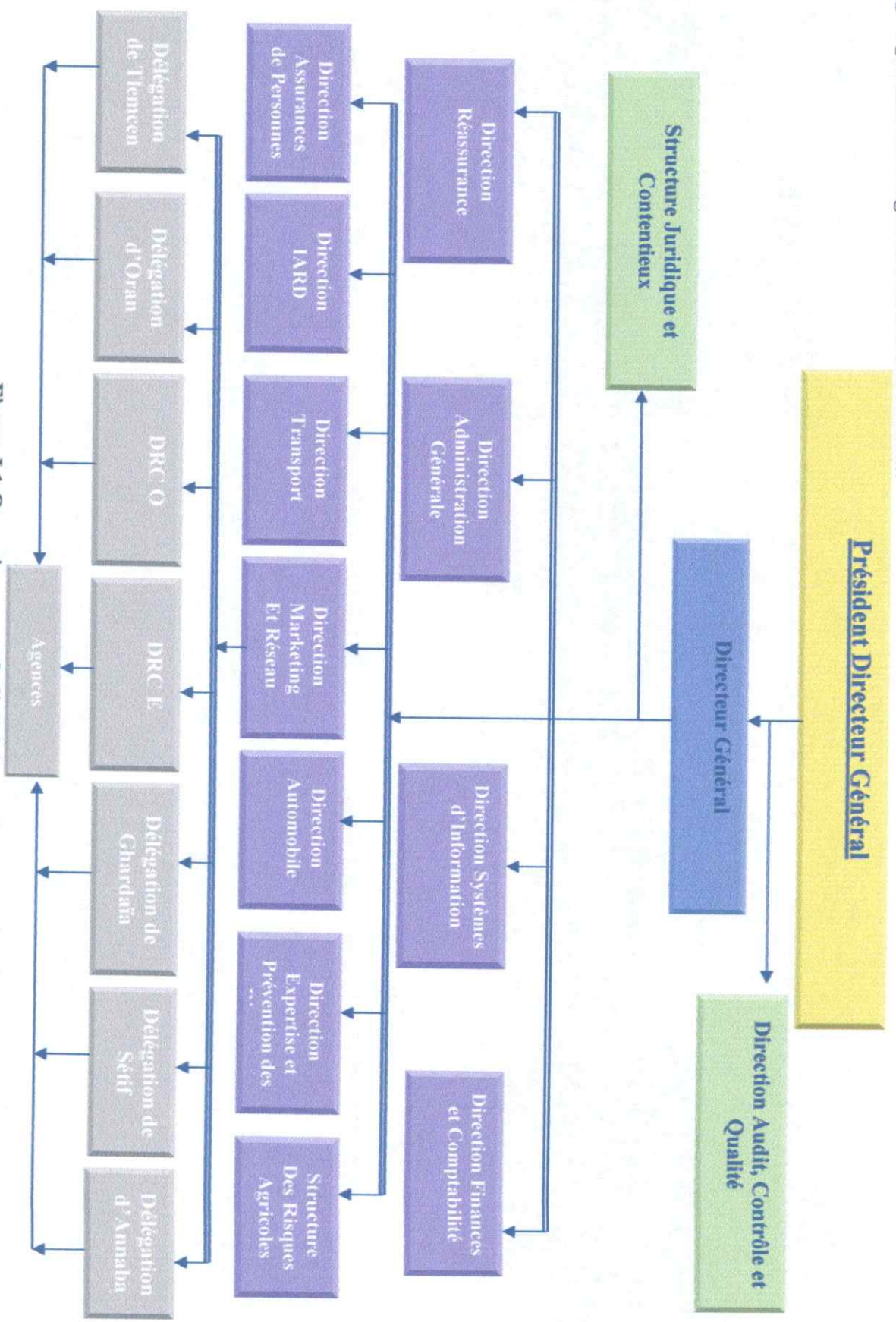


Figure I.1 Organigramme de l'algérienne des assurances2

I.4. Organisation des délégations de l'algérienne des assurances2a:

L'organisation de la 2a est établie selon l'organigramme horizontal. Son but est d'avoir la minimum distance entre la direction général et les délégations régionales la et avoir une ligne uniforme et réduite concrétiser par une prestation effectuée dans des délais très courts.

La 2a est organisé en 6 délégations régionales:

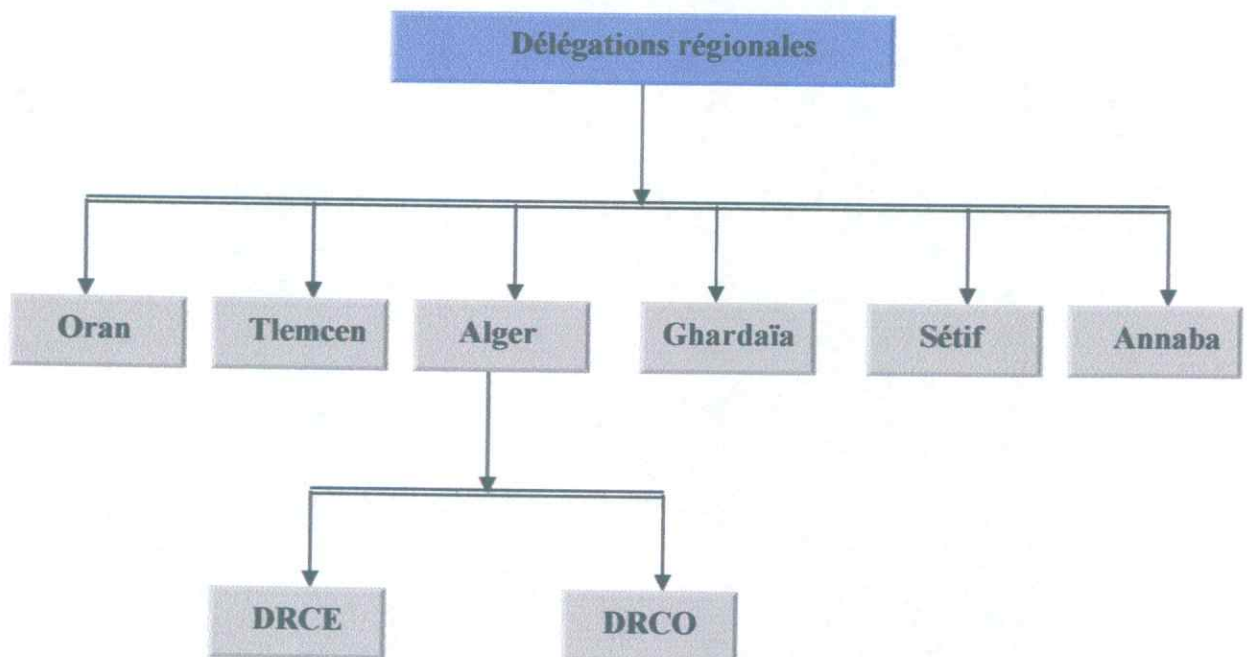


Figure I.2 Organigramme des délégations l'algérienne des assurances2

II. Présentation de la structure d'accueil:

Les structures concernées par notre étude sont les agences, chaque agence est dirigée par un chef d'agence sous l'autorité du directeur régional.

II.1 Missions et objectifs:

Les agences sont chargées de la gestion de la production, elles ont pour rôle:

- Rechercher des clients cibles.
- Etablissement et gestion des d'assurances.
- Etablissement des statistiques.
- Prospection de la clientèle de proximité (particuliers, ménage, profession libérale).
- Présentations de qualités à la clientèle.
- Prises en charges des sinistres (ouverture des dossiers, expertisions, indemnisation).

II.2. l'organisation des agences:

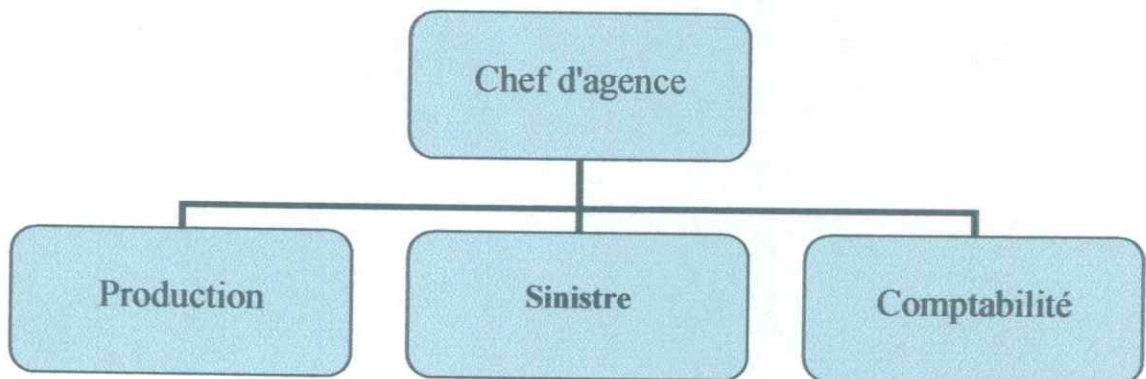


Figure I.3 : L'organisation des agences

Chapitre II: l'étude de l'existant

1.Introduction:

L'étude de l'existant est une étape très importante à toute application de conception, elle nous permet d'aboutir à la réalité de l'organisation.

L'analyse arrive à une maîtrise totale de l'ensemble des traitements des procédures ainsi que les documents et les fichiers. Nous avons eu recours à la collecte des informations à travers des interviews avec les responsables ainsi que des observations faites sur place, pour aboutir enfin aux critiques et suggestions qui seront la base de notre étude conceptuelle. Cette étude permet d'établir une image aussi précise que possible de l'organisation.

II.Lexique:**II.1 Définition :**

L'assurance groupe c'est un produit d'assurance appartient à l'assurance des personnes, l'assurances groupes peut être définie comme l'assurance d'un ensemble de personnes présentant des caractéristiques communes et relevant des même conditions techniques pour l'évolution des risques assurés.

L'assurance de personnes au point de vue juridique (ordonnance 95-07 du 25 Janvier1995) se présente comme une convention, un contrat entre l'assuré et l'assureur. De la définition juridique de cette branche d'assurance, ressort un ensemble de personnes intervenant au contrat, dont l'assureur, l'assuré, le souscripteur et le bénéficiaire.

II.2 Autre définitions:**1. L'assureur :**

C'est une personne morale (compagnie d'assurance) qui élabore le contrat, prend des engagements vis à vis de l'assuré ou du souscripteur et gère le contrat.

2. le groupe assuré :

Ensemble d'individus appartenant à une entreprise ou tout organisme constitué pour une raison autre que la souscription des contrats: salariés d'une entreprise, membres d'une association, bénéficiaire de prêts auprès d'une banque, adhérents d'une mutuelle.

3. Le souscripteur :

C'est une personne physique ou morale qui s'engage vis à vis de l'assureur sur la base des conditions de l'assurance stipulées dans le contrat.
Elle signe le contrat, paie les primes et gère le contrat.

Ne peuvent jouer le rôle de souscripteurs que les personnes physiques ou morales autorisées par la loi (articles 68 et 69 de l'ordonnance 95-07).

4.Le bénéficiaire :

C'est une personne physique ou morale qui recueille les avantages stipulés dans le contrat d'assurance conformément aux dispositions de la loi.

Remarque :

Selon les types de contrats, les qualités d'assuré, du souscripteur et du bénéficiaire peuvent se trouver réunies.

Bien souvent, dans les assurances en cas de vie, mixtes, incapacité ou invalidité, l'assuré peut être en même temps:

- le souscripteur;
- le bénéficiaire;
- le souscripteur et le bénéficiaire.

Tandis que dans les assurances en cas de Décès, l'assuré peut se trouver en même temps souscripteur, mais en aucun cas bénéficiaire.

5. Police:

La police d'assurance est l'imprimé qui constate l'existence du contrat d'assurance.

L'ordonnance 95-07 du 25 Janvier 1995, dans son article 7, précise les mentions qui doivent y figurées obligatoirement.

Ces mentions sont complétées en assurance de personnes par l'article 70 de la même ordonnance.

Le premier prescrit que la police doit contenir obligatoirement, outre les signatures des parties, les dispositions ci-après:

Les noms et domiciles des parties contractantes;

La personne assurée;

La nature des risques garantis;

La date de souscription;

La date d'effet et la durée du contrat;

Le montant de la garantie;

Le montant de la prime ou cotisation d'assurance.

Le second énonce que la police d'assurance de personnes doit contenir:

Les noms, prénom(s) et date(s) de naissance de(s) l'assuré(s);

Les noms et prénoms des bénéficiaires s'ils sont déterminés;

L'événement ou le terme duquel dépend l'exigibilité des sommes assurées;

Les dispositions en matière de réduction, de rachat et les conditions d'application, conformément aux articles 84, 85 et 90 de la même ordonnance.

Pratiquement la police d'assurance se présente comme un document divisé en deux parties où doivent y figurer toutes ces indications: les conditions générales et particulières.

Si une ou plusieurs indications citées ci-dessus ne figurent pas dans les imprimés, le rédacteur du contrat doit la ou les compléter dans les conditions particulières.

5.1 Conditions générales :

Le document des conditions générales est imprimé d'avance et correspond à la combinaison d'assurance choisie par l'assuré ou par le souscripteur.

Selon le type de contrat d'assurance, une série de rubriques est consacrée, dont voici les principales:

- **Prise d'effet du contrat;**
- **Paieiment de la prime;**
- **Conséquences du non paiement de prime;**
- **Risques exclus;**
- **Différentes opérations auxquelles le contractant pourra se livrer en cours de contrat: transformation, réduction, rachat, avance, réversion, etc... .**
- **Conditions de règlement en cas de réalisation du risque assuré.**

5.2 Conditions particulières :

Bien que le document des conditions particulières soit imprimé d'avance, il peut faire l'objet de complément de clauses particulières aux conditions générales.

Sous réserve de l'article 6 de l'ordonnance 95-07, après accord entre les parties contractantes, une ou plusieurs clauses peuvent être introduites pour déroger aux conditions générales du contrat.

L'assurance de groupe est toujours souscrite par une personne morale : société, entreprise, association, etc..

Quels sont les risques qui peuvent être couverts ?

Risques qui dépendent de la durée de vie humaine :

Décès ou (et) vie.

Incapacité de travail résultant de maladie ou d'accident :

Incapacité Permanente et Totale ; I.P.T.

Incapacité Permanente Partielle ; L.P.P.
Incapacité Temporaire de Travail. I.T.T.

6. La garantie:

Est déterminée suivant un critère objectif pour tous: salaire, situation de famille, catégorie professionnelle, ancienneté,...

A. Garantie de base:

6.1 Décès et Invalidité Absolue et Définitive:

On entend par invalidité absolue et définitive, lorsque l'Assuré, est réputé depuis plus de douze mois dans un état d'invalidité irréversible, le mettant dans l'impossibilité de se livrer à un travail ou à une occupation quelconque lui rapportant gain ou profit et nécessitant l'assistance d'une tierce personne pour accomplir les actes ordinaires de la vie (au sens de la 3^{ème} catégorie des invalides de la sécurité sociale).

B. Garantie complémentaire:

6.2 exonération de paiement des primes:

En cas d'Incapacité Temporaire de Travail d'un agent de la contractante plus de trois mois (03 mois) suite à une maladie ou un accident, ou bien, s'il est déclaré en état d'Incapacité Permanente Partielle plus de Cinquante pour cent (50 %) suite à une maladie ou un accident, il continu à bénéficier des garanties du présent contrat sans paiement de prime à partir du quatrième mois de cessation d'activité.

6.3 Accident de circulation:

En entent par accident de circulation, tout accident causer sur un voie public ou privée ouverte au public par l'action d'un véhicule, que ce véhicule transporte ou non l'assuré, et le transporte (soit conducteur ou passager simple), est également considéré comme accident de la circulation, tout accident de l'usage fait par l'assuré, à titre de simple passager, de tout moyen de transport en commun terrestre, fluvial, maritime, ou aérien sur ligne commerciale régulière.

6.4 Indemnités forfaitaire:

Des Indemnités forfaitaires sont alloués à l'occasion de la survenance d'un des évènements ci-après:

Naissance, mariage, décès d'un prestataire, départ retraite,...

6.5 Maladie et soins annexes:

___ Cette garantie a pour objet de couvrir les personnes définies ci-dessous contre les risques de maladies, d'accidents, de maternité, etc., lorsque les soins sont dispensés par des praticiens diplômés et reconnus comme tels par le ministère de la santé publique.

En aucun cas le remboursement de la Sécurité Sociale ou tout autre organisme de prévoyance, cumulé au remboursement de la 2a ne doit dépasser les débours réels.

6.6 Incapacité temporaire: Lorsque par suite de maladie ou accident, un assuré est obligé de cesser son travail, l'assurance garantie le paiement d'une suite ininterrompue de journée d'incapacité totale, à partir de la date de la cessation du travail.

6.7 Rentes éducation:

Au Décès de l'assuré, la 2A. S'engage à verser en plus du capital Décès, une rente temporaire à chacun des enfants mineurs de l'assuré à charge au sens des allocations familiales, dont le montant est fixé aux conditions particulières

Cette rente est payable trimestrielle ment à terme échu, et elle cesse d'être due de plein droit lorsque l'enfant aura atteint son 18ème anniversaire.

6.8-Incapacité permanente et partielle (IPP) :

Lorsque l'assuré par suite de maladie ou d'accident est atteint d'une incapacité permanente de travail (Art.27 des conditions générales), c'est-à dire réduisant sa capacité de tirer un revenu de son travail, la « 2A » lui versera une rente d'invalidité jusqu'à 60 ans de la manière suivante:

* Si le taux d'incapacité est égal ou supérieur à 50 % et que l'assuré est classé par la S.S. dans la 2eme catégorie.

* Si le taux d'incapacité est égal ou supérieur à 50 % et que l'assuré est classé par la S.S. dans la 2Ème catégorie des invalides , la rente annuelle est égale à 30% du salaire de référence.

* Si le taux d'incapacité est inférieur à 50 % et que l'assuré ne bénéficie d'aucune rente au titre des accidents du travail ou d'une maladie professionnelle, la 2A lui versera une rente après avoir réduit le taux d'incapacité d'une franchise absolue de 30 %

* Cette rente est payable trimestriellement à terme échu et elle est servie aussi longtemps que l'assuré justifie de son état d'invalidité.

7. Avenant : toute modification attribué à un contrat (renouvellement, suppression garantie, suppression adhérent, ajout adhérent, ajout garantie ;

7.1 Avenant de reconduction : renouvellement

7.2 Avenant d'adjonction : ajout

7.3 Avenant d'annulation : suppression

III présentation de la situation actuelle:**III.1 Le flux d'information:**

C'est une vue globale sur l'échange des informations dans le système actuel. On représente le flux d'information par un graphe général où les nœuds représentent les acteurs et les arcs représentent la circulation des informations entre les acteurs.

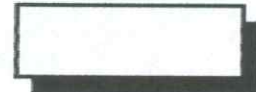
La codification des acteurs:

Code	Désignation
CL	client
CA	Chef d'agence
AP	Direction assurances des personnes
PRD	Service production
SNR	Service sinistre
DR	Délégation régionale
CMP	Service comptabilité

1. Acteur interne au champ d'étude:



2. Acteur interne à l'organisme, externe au champ d'étude:



3. Acteur externe à l'organisme:



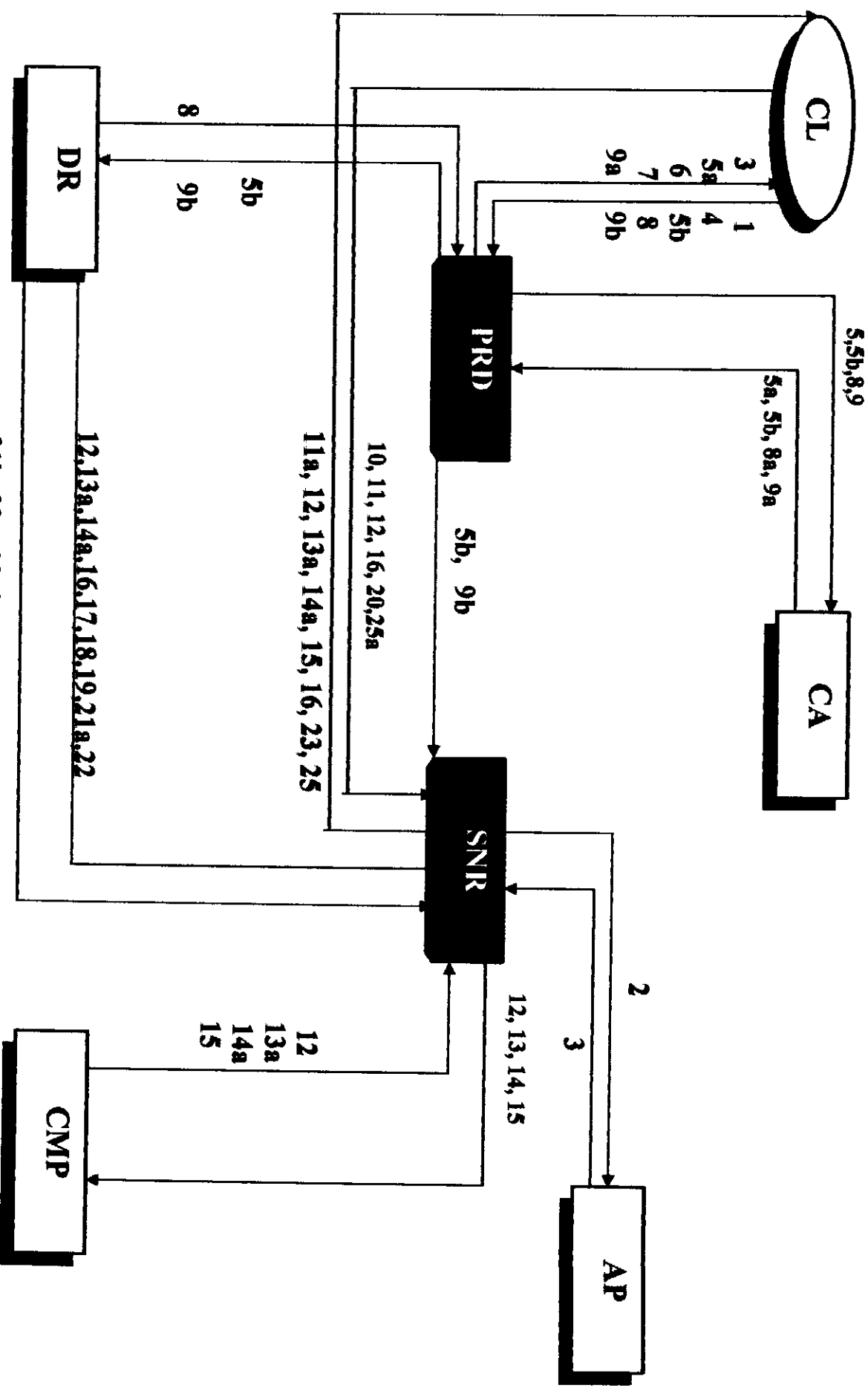


Figure II.1: graphe générale de flux d'information

♦ Description du graphe de flux d'information:

Code	désignation
1	Demande du CL
2	Demande orale de tarification
3	Tarification verbale
4	Réponse du CL
5	Etablissement du contrat
5a	Contrat signé par le CA
5b	Contrat signé par le CL
6	Demande individuelle d'adhésion
7	Feuille de maladies
8	Demande du CL ou de la DR
8a	Demande du CL ou de la DR accepté
9	Etablissement d'un avenant
9a	Avenant signé par le CA
9b	Avenant signé par le CL
10	Dossiers des sinistres
11	Bordereau d'envoi
11a	Bordereau signé par le service PRD
12	Bordereau des remboursements globaux
13	Ordre de paiement
13a	Ordre de paiement signé par le CMP
14	Quittance de règlement
14a	Quittance de règlement signé par le CMP
15	Chèque
16	Fiche de décompte de remboursement
17	Bordereau de sinistres déclarés
18	Bordereau de sinistres réglés
19	Bordereau d'envoi
20	Dossier du CL (sinistre)
21	20+16
21a	21 signé par le CA
21b	21a signé par la DR
22	11
22a	22 signé par la DR
23	15
24	14
25	13
25a	25 signé par le CL

Tableau II.1: Description du graphe de flux d'information

III.2 Etude des postes de travail, les documents et les registres:

♦ La codification des documents et registres:

code	désignation
DCL	Demande du client
DIA	Demande individuelle d'adhésion
CAG	Contrat d'assurance groupe
AVE	Avenant
BRG	Bordereau des remboursements globaux
OP	Ordre de paiement
BE	Bordereau d'envoi
BSD	Bordereau des sinistres déclarés
BSR	Bordereau des sinistres réglés
FDR	Fiche de décompte des remboursements
FM	Feuille de maladie
FDDIR	Fiche de décompte décès, invalidité, rente éducation
QR	Quittance de règlement
RSD	Registres des sinistres déclarés
RSR	Registres des sinistres réglés
A	Alphabétique
AN	Alphanumérique
D	Date

Tableau IL2: Description de la codifications des documents et registres

III.2.1 Etude des postes de travail:

Le poste de travail est un ensemble défini de tâche, de voix, de responsabilité, qui constitue le travail habituelle du personne. Cette étude permet de connaître la structure de l'organisme et de comprendre les procédures administratives utilisées au niveau de chaque poste de champ d'étude.

A) Service production:

Fiche de fonction N=°1		
Désignation du poste: Service production		
Lieu du poste: L'agence		
Responsable du poste: Chef de service production		
Effectif: 01		
01-Tâche à accomplir		
1. réception des demandes de souscription ou d'avenant des clients 2. réception et transmission des tarifications 3. établissement des contrats 4. établissement des DIA et FM 5. établissement des avenants 6. classement des avenant, DIA et une copie du contrat 7. envoi des FM, DIA, le contrat et les avenants		
02-Document provenant à ce poste		
2.1-Documents provenant à ce poste		
Documents	Emetteur	Observation
1. Demande de souscription	CL CL	/ /
2. Demande avenant	DR	La demande a été faite par appel téléphonique
3. Tarification		
2.2-Documents remplis et diffusé de ce poste		
Documents	Nombre d'exemplaire	Destination
1. Contrat	04	CL et DR
2. DIA	03	CL et service sinistre
3. Avenant	03	DR et service sinistre
4. FM	Variable	CL

Tableau II.3: Description de la fiche de fonction N=°1

B) Service sinistre:

Fiche de fonction N=°2		
Désignation du poste: Service sinistre		
Lieu du poste: L'agence		
Responsable du poste: Chef de service sinistre		
Effectif: 01		
01-Tâche à accomplir		
1. gestion des sinistres 2. réception des dossiers des clients 3. établissement des FD, BRG, BE, OP, QR 4. envoi les chèque aux clients et classement d'une copie 5. classement et envoi des BRG, OP, QR, FDD aux clients 6. classement et envoi des BSD, BSR, BRG, FDD, BE, OR, OP aux DR		
02-Document provenant à ce poste		
2.1-Documents provenant à ce poste		
Documents	Emetteur	Observation
1. Dossiers du CL	CL	/
2. Chèque	CMP / DR	/
2.2-Documents remplis et diffusé de ce poste		
Documents	Nombre d'exemplaire	Destination
OP	Variable	CL
QR	3	CL /DR
BSD	3	DR
BSR	3	DR
BRG	3	CL / DR
FDD	3	CL / DR
BE	2	DR

Tableau II.4: Description de la fiche de fonction N=°2

II.2.2 Etude des documents:**Contrat Assurance Groupe**

Fiche document n=°1		
Code: CAG Désignation: Contrat d'assurance groupe Rôle: Etablir une convention entre l'assuré et l'assureur origine: Service production destination: assuré/délégation/agence Format: A4 Papier: normal Nombre d'exemplaire: 03 Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Nom de l'assuré	A	-
Adresse de l'assuré	AN	-
Numéro de police	N	-
Condition particulière	AN	-
Date début de contrat	D	JJ/MM/AAAA
Date fin du contrat	D	JJ/MM/AAAA
Signature de l'assuré	-	Cachet et visa
Signature de l'assureur	-	Cachet et visa

Tableau II.4: Description de la fiche de document N=°1

Avenant

Fiche document n=°2		
Code: AVE		
Désignation: Avenant		
Rôle: déclare tout changement que peut faire le client sur le contrat		
origine: Service production		
destination: assuré/délégation/agence		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 03		
Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Numéro de police	N	JJ/MM/AAAA
Effet de l'avenant	D	JJ/MM/AAAA
Echéance de la police	D	
Nom contractant	A	
Adresse	AN	
Adhérent	A	
Adresse de l'adhérent	AN	
Type avenant	A	
Numéro de la police	N	
Numéro de l'avenant	N	
Article	AN	
<u>Détail:</u>	-	JJ/MM/AAAA
Période	D	
Prime nette	N	
Accessoire	N	
Droit de timbres	N	
Totale à payer	N	JJ/MM/AAAA
Date de l'avenant	D	
Adhérent		Cachet et visa
La société		Cachet et visa

Tableau II.5: Description de la fiche de document N=°2

Bordereau des sinistres déclaré

Fiche document n=°3		
Code: BSD		
Désignation: Bordereau des sinistres déclaré		
Rôle: déclaration des sinistres		
origine: Service sinistres		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 01		
Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Date	D	JJ/MM/AAAA
Code agence	AN	
Période	D	-
Code branche	N	-JJ/MM/AAAA
Numéro de sinistre	N	
Numéro de police	N	-
Effet	D	JJ/MM/AAAA
Echéance	D	JJ/MM/AAAA
Date de sinistre	D	JJ/MM/AAAA
Nom Assuré	A	-
Code agence	AN	-
Numéro séquentiel	N	-
<u>Total réserve à payer:</u>		
Montant	N	
Date	D	JJ/MM/AAAA
Totalisation produit:	N	-

Tableau II.6: Description de la fiche de document N=°3

Bordereau des sinistres réglé

Fiche document n=°4		
Code: BSR Désignation: Bordereau des Sinistres Réglé Rôle: déclaration des sinistres Réglé origine: Service sinistres Format: A4 Papier: normal Nombre d'exemplaire: 01 Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Date	D	JJ/MM/AAAA
Code agence	AN	-
Période	D	JJ/MM/AAAA
Code branche	N	-
Numéro de sinistre	N	-
Numéro de police	N	-
Effet	D	JJ/MM/AAAA
Echéance	D	JJ/MM/AAAA
Date de sinistre	D	JJ/MM/AAAA
Nom Assuré	A	-
Bénéficiaire	A	-
Règlement:		-
Type garanti	A	
Date	D	JJ/MM/AAAA
Montant	N	-
Total	N	-

Tableau II.7: Description de la fiche de document N=°4

Ordre de paiement

Fiche document n=°5		
Code:OP		
Désignation:Ordre de paiement		
Rôle:ordre pour paiement		
origine: Service sinistres et comptabilité		
Format:A4		
Papier: normal		
Nombre d'exemplaire:01		
Verso:non		
Analyse de contenu		
Rubrique	Type	Observation
Numéro d'ordre de paiement	N	JJ/MM/AAAA
Montant	N	
Nature de l'opération	A	-
Numéro de police	AN	-JJ/MM/AAAA
Numéro de sinistre	N	
Numéro d'avenant	AN	-
Ordonnateur	A	JJ/MM/AAAA
Nom de l'assuré	A	JJ/MM/AAAA
Bénéficiaire	A	JJ/MM/AAAA
Montant à régler	A	-
Visa de la structure	-	-
Date	D	-
Nature de règlement	AN	
Visa	-	Signature
Partie prenante	AN	-
Pour acquis	AN	-
Compte à débiter	N	-
Compte à créditer	N	-
Montant débit	N	-
Montant crédit	N	-
Visa de la structure	-	Signature

Tableau II.8: Description de la fiche de document N=°5

Quittance de règlement

Fiche document n=°6		
Code: QR Désignation: Quittance de règlement Rôle: Quittance pour règlement origine: Service sinistre et comptabilité Format: A4 Papier: normal Nombre d'exemplaire: 01 Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
délégation	A	
Code agence	AN	-
Nom assuré	A	-
Numéro de police	AN	-
Numéro de sinistre	N	-
Date d'accident	D	
Type garanti	A	JJ/MM/AAAA
Montant à régler	N	-
Article	A	-
Date de règlement	D	-
<u>Payement:</u>		JJ/MM/AAAA
Numéro de chèque	N	
Nom de banque	A	-
Numéro d'agence de la banque	N	-
Signature du chef d'agence 2a	-	-
		Cachet et visa
		-
		Signature

Tableau II.9: Description de la fiche de document N=°6

Demande du client

Fiche document n=°7		
Code: DCL		
Désignation: Demande du client		
Rôle: demande d'adhésion dans l'assurance groupe		
origine: le client		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 01		
Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Nom assuré	A	-
Adresse	AN	-
Capital social	N	-
Nombre d'effectif	N	-
Masse salariale	N	-
Objet	A	-
Contenu de la demande	AN	-
Date	D	JJ/MM/AAAA
Signature de client	-	Cachet et visa

Tableau II.10: Description de la fiche de document N=°7

Bordereau d'envoi

Fiche document n=°8		
Code: BE		
Désignation: Bordereau d'envoi		
Rôle: déclaration d'envoi d'un document		
origine: le service sinistre		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 01		
Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Date	D	JJ/MM/AAAA
Code agence	AN	-
Nom agence	A	-
Désignation des pièces	AN	-
Nombre de pièces	N	-
Observation	AN	-
Signature	-	Cachet et visa

Tableau II.11: Description de la fiche de document N=°8

Feuille de maladie

Fiche document n=°9		
Code: FM Désignation: Feuille de maladie Rôle: permet à l'assuré de faire rembourser origine: service sinistre destination: agence Format: A4 Papier: normal Nombre d'exemplaire: N Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Nom de la délégation	A	
Code agence	AN	-
Dénomination sociale du souscripteur	AN	-
Adresse	AN	-
Numéro de police	AN	-
Nom et prénom de l'assuré	A	-
Nom et prénom du malade	A	-
Adresse du malade	AN	
Date des actes médicaux	D	JJ/MM/AAAA
Nature de l'acte	AN	-
Cachet et visa du praticien		
Montant des horaires à payer	-	-
Montant de la facture	N	-
Cachet du pharmacien	N	-
Date d'entrée	-	-
Date de sortie	D	JJ/MM/AAAA
Nature de l'acte	D	JJ/MM/AAAA
Montant des horaires à payer	AN	-
Date de sortie	N	-
Date	D	JJ/MM/AAAA
lieu	D	JJ/MM/AAAA
	A	-

Tableau IL.12Description de la fiche de document N=°9

Fiche de décompte de remboursement

Fiche document n°10		
Code: FDR		
Désignation: Fiche de décompte de remboursement		
Rôle: détaille les décomptes (décès/IAD)		
origine: service sinistre		
destination: assuré/délégation/agence		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 03		
Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Nom de la délégation	A	-
Code agence	AN	-
nom du souscripteur	A	-
date début du contrat	D	JJ/MM/AAAA
date fin du contrat	D	JJ/MM/AAAA
Numéro de police	AN	-
Numéro du sinistre	AN	-
Nom et prénom de l'assuré	A	-
Situation familiale	A	-
Nombre d'enfant	N	-
Numéro d'adhésion	AN	-
Cause de décès ou invalidité	A	-
Montant total des 12 mois de salaire	N	-
Capitale de base	N	-
Majoration par enfant à charge	N	-
Total à payer	N	-
Primes restantes dues	N	-
Position de l'assuré	N	-
Capital constitutif pour rente éducation	N	-
Montant globale du sinistre	N	-
Bénéficiaires	A	-
Visa du chef d'agence	-	Cachet et signature
Visa du directeur de la délégation	-	Cachet et signature
Visa du directeur technique du siège	-	Cachet et signature
Visa de la direction générale	-	Cachet et signature
La somme à payer	N	-
Date	D	JJ/MM/AAAA

Tableau II.13 Description de la fiche de document N°10

Bordereau des remboursements globaux

Fiche document n°11		
Code: BRG		
Désignation: Bordereau des remboursements globaux		
Rôle: déclaration de tous les remboursement		
origine: service sinistre		
destination: assuré/délégation/agence		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 01		
Verso: non		
Analyse de contenu		
Rubrique	Type	Observation
Moi	D	JJ/MM/AAAA
Nom de l'assuré	A	-
Nom et prénom de l'adhérent	A	-
Numéro du dossier	N	-
Nombre de décompte	N	-
Montant	N	-
Observation	A	-
Total	N	-
Date	D	JJ/MM/AAAA
Lieu	A	-
Signature du correspondant	-	-
Signature de l'assureur	-	-

Tableau IL14: Description de la fiche de document N°11

Demande Individuelle d'adhésion

Fiche document n°12		
Code: DIA		
Désignation: Demande Individuelle d'adhésion		
Rôle: demande Individuelle d'adhésion dans l'assurance groupe		
origine: assuré/service production		
Format: A4		
Papier: normal		
Nombre d'exemplaire: 03		
Verso: oui		
Analyse de contenu		
Rubrique	Type	Observation
Numéro de police	AN	-
Numéro du certificat	AN	-
Effet de l'adhésion	D	JJ/MM/AAAA
Nom ou raison social	A	-
Adresse	AN	-
Numéro de la sécurité sociale	AN	-
Nom de l'assuré	A	-
Prénom	A	-
Date de naissance	D	JJ/MM/AAAA
Adresse	AN	-
Situation familiale	A	-
Nom et prénom du conjoint	A	-
Date de naissance du conjoint	D	JJ/MM/AAAA
Nombre d'enfant à charge	N	-
Date de naissance de l'enfant	D	JJ/MM/AAAA
Nom de la contractante	A	-
Montant en chiffre	N	-
Montant en lettre	N	-
Nom et prénom de bénéficiaire en cas décès	A	-
Lieu éventuel de parenté	A	-
Questionnaire	A	-
Cachet du contractant	-	Cachet et visa
Lieu de la signature	A	-
Date de la signature	D	JJ/MM/AAAA
Signature de l'adhérent	-	-

Tableau II.15: Description de la fiche de document N°12

III.2.3 Etude des registres

Il existe deux types de registres:

1. **registres de production**: est divisé sur deux parties:

1.1 **police**: contient les information du document police classé par mois.

1.2. **Avenant**: contient les information du document avenant classé par mois.

2. **registres des sinistres**: est divisé sur deux parties:

2.1 **Sinistres déclarés**: contient les information du document bordereau des sinistres déclaré classé par mois.

2.2 **Sinistres réglés**: contient les information du document bordereau des sinistres réglés classé par mois.

III.3 Etude de la codification :

III.3.1 Définition:

Un code est une forme abrégée qui identifié un objet sans ambiguïté.

La codification c'est faire passer l'information d'un objet de sa forme naturelle à une forme symbolique significatif.

Un code doit aussi permettre des extensions des insertions.

III.3.2 règle:

Une codification doit respecter les contraintes suivantes:

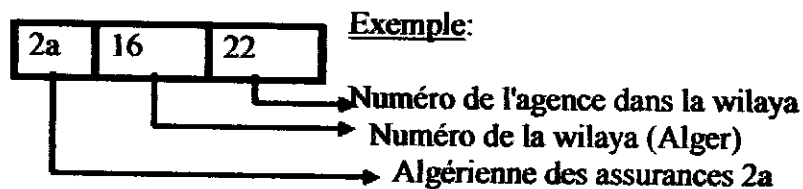
- ◆ Un code désigne un et un seul objet.
- ◆ Un code doit être le plus significatif possible.
- ◆ Avoir une codification facile et simple.
- ◆ La codification doit permettre la réduction de l'espace et le gain du temps.

III.3.3 Codification existante:

Voici la codification existante:

✓ Agence :

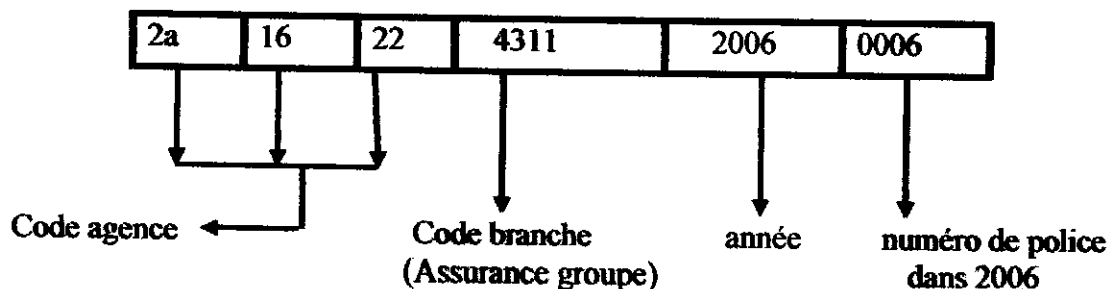
Code alphanumérique sur 6 positions



✓ police:

Code alphanumérique sur 18 positions

Exemple:



✓ Avenant :

Code alphanumérique sur 20 positions, a le même code que la police en ajoutant un numéro sur deux positions signifie le type d'avenant

Exemple:

2a	16	07	11	4311	2005	0022
----	----	----	----	------	------	------

Code signifie le type d'avenant
(11:avenant de reconduction)

✓ Adhésion:

Code alphanumérique sur 2 positions.

Exemple: A1

A: adhésion

1: numéro d'adhésion

IV. Diagnostic:

L'étude de l'existant à nous donner une vue globale du déroulement de travail au service de l'assurance groupe, l'étude met en évidence les insuffisances et permet d'avoir les critiques suivantes:

- Manque d'effectif (la même personne s'occupe du service production et service sinistre).
- La codification utilisée n'est pas uniforme.
- L'existence des rubriques non utilisée dans les documents.
- La durée du transfert des informations entre les agences et les délégations se fait à la fin de chaque moi.

Pour compléter les insuffisances dans le système actuel, on a pensé à des améliorations qui se présentent comme suit:

- Simplifier la codification.
- Eliminer les rubriques non utilisée dans les documents.
- Il faut penser à introduire la notion : exécution des taches à temps réel pour assurer le transfert des taches entre les agences et les délégations à son exécution.

Chapitre III: Conception

I. Introduction :

L'étape précédente dans le cycle de réalisation de notre nouveau système nous a permis de bien comprendre le fonctionnement de l'ancien système et définir les failles et les insuffisances de ce système. Pour dépasser ses obstacles on a arrivé à proposer une solution, mais avant d'implémenter cette solution il faut passer par l'étape de conception, cette étape indispensable permet de donner une réflexion uniforme pour l'implémentation de notre nouveau système informatique en définissant les différents modèles d'application. Pour cela on utilise la modélisation UML.

II. Présentation de l'UML:

II.1.Introduction :

Le génie logiciel et la méthodologie s'efforcent de couvrir tous les aspects de la vie du logiciel. Issus de l'expérience des développeurs, concepteurs et chefs de projets, ils sont en constante évolution, parallèlement à l'évolution des techniques informatiques et du savoir-faire des équipes.

Comme toutes les tentatives de mise à plat d'une expérience et d'un savoir-faire, les méthodologies ont parfois souffert d'une formalisation excessive, imposant aux développeurs des contraintes parfois contre-productives sur leur façon de travailler.

Avec la mise en commun de l'expérience et la maturation des savoir-faire, on voit se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes.

UML, qui se veut un instrument de capitalisation des savoir-faire puisqu'il propose un langage qui soit commun à tous les experts du logiciel, va dans le sens de cet assouplissement des contraintes méthodologiques.

II.2.la modélisation :

Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer.

Il permet de:

- De visualiser le système comme il est ou comme il devrait l'être.
- De valider le modèle vis à vis des clients
- De spécifier les structures de données et le comportement du système.
- De fournir un guide pour la construction du système.
- De documenter le système et les décisions prises.

II.3.Apport de la modélisation objet :

- Plus grande indépendance du modèle par rapport aux fonctionnalités demandées.
 - Des fonctionnalités peuvent être rajoutées ou modifiées, le modèle objet ne change pas.
- Plus proche du monde réel.

II.4.Définition de l'UML:

UML signifie Unified Modeling Language. La justification de chacun de ces mots nous servira de fil conducteur pour cette présentation.

1. Unified : historique des méthodes de conception :

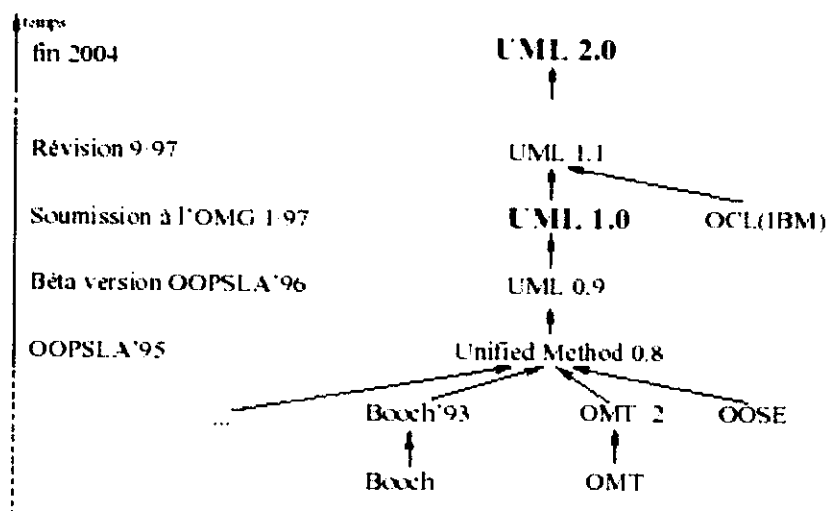


Figure.III. 1 – Historique de la constitution d'UML

À chacune des différentes phases de la conception d'un logiciel correspondent des Problèmes ou des contraintes différentes. Naturellement, ces niveaux ont fait l'objet de recherches méthodologiques considérables depuis les années 80. Il en résulte que de nombreuses méthodes de développement ou d'analyse de logiciel ont vu le jour, Chacune plus ou moins spécialisée ou adaptée à une démarche particulière, voire à un secteur industriel particulier (bases de données, matériel embarqué, ...) . Celles-ci ayant été développées indépendamment les unes des autres, elles sont souvent partiellement redondantes ou incompatibles entre elles lorsqu'elles font appel à des notations ou des terminologies différentes, voire à des faux amis.

De plus, à chaque méthode correspondent un ou plusieurs moyens (plus ou moins formel) de représentation des résultats. Celui-ci peut être graphique (diagramme synoptique, plan physique d'un réseau, organigramme) ou textuel (expression d'un besoin en langage naturel, jusqu'au listing du code source). Dans les années 90, un certain nombre de méthodes orientées objets ont émergé, en particulier les méthodes :

- OMT de James RUMBAUGH ,
- BOOCH de Grady BOOCH ,
- OOSE (*Object Oriented Software Engineering*) de Ivar JACOBSON à qui l'on doit les Use cases .

En 1994, on recensait plus de 50 méthodologies orientées objets. C'est dans le but de remédier à cette dispersion que les « poids-lourds » de la méthodologie orientée objets ont entrepris de se regrouper autour d'un standard.

En octobre 1994, Grady Booch et James Rumbaugh se sont réunis au sein de la société RATIONAL dans le but de travailler à l'élaboration d'une méthode commune qui intègre les avantages de l'ensemble des méthodes reconnues, en corrigeant les défauts et en comblant les déficits. Lors de OOPSLA'95 (*Object Oriented Programming Systems, Languages and Applications*, la grande conférence de la programmation orientée objets), ils présentent UNIFIED METHOD V0.8. En 1996, Ivar Jacobson les rejoint.

Leurs travaux ne visent plus à constituer une **méthodologie**, mais un **langage**.

Leur initiative a été soutenue par de nombreuses sociétés, que ce soit des sociétés de développement (dont Microsoft, Oracle, Hewlet-Packard, IBM – qui a apporté son langage de contraintes OCL –, ...) ou des sociétés de conception d'ateliers logiciels.

Un projet a été déposé en janvier 1997 à l'OMG en vue de la normalisation d'un langage de modélisation. Après amendement, celui-ci a été accepté en novembre 97 par l'OMG sous la référence UML-1.1. La version UML-2.0 est annoncée pour la fin 2004.

2. Modeling : analyse et conception :

Une bonne méthodologie de réalisation de logiciels suppose une bonne maîtrise de la distinction entre l'analyse et la conception. Le lecteur verra qu'en pratique, le respect d'une distinction entre des phases d'analyse et de conceptions rigoureusement indépendantes n'est pas tenable, mais il est important d'avoir en tête la différence lorsqu'on s'apprête à réaliser un logiciel. Encore une fois, il est important de garder à l'esprit qu'UML n'offre pas une méthodologie pour l'analyse et la conception, mais un langage qui permet d'exprimer le résultat de ces phases.

Du point de vue des notations employées en UML, les différences entre l'analyse et la conception se traduisent avant tout par des différences de niveau de détail dans les diagrammes utilisés. On peut ainsi noter les différences suivantes :

- Dans un diagramme de classes d'analyse, les seules classes qui apparaissent servent à décrire des objets concrets du domaine modélisé.
- Dans un diagramme de classes de conception, par opposition, on trouve aussi toutes les classes utilitaires destinées à assurer le fonctionnement du logiciel.
- Dans un diagramme de classes d'analyse, on peut se contenter de faire apparaître juste la dénomination des classes, avec parfois le nom de quelques attributs et méthodes quand ceux-ci découlent naturellement du domaine modélisé.
- Dans un diagramme de classes de conception, par opposition, tous les attributs et toutes les méthodes doivent apparaître de façon détaillée, avec tous les types de paramètres et les types de retour.
- Dans un diagramme de séquence d'analyse, les communications entre les principaux objets sont écrits sous forme textuelle, sans se soucier de la forme que prendront ces échanges lors de la réalisation du logiciel. Dans un diagramme de séquence de conception, par opposition, les échanges entre classes figurent sous la forme d'appels de méthodes dont les signatures sont totalement explicitées.

3. Language : méthodologie ou langage de modélisation ?

Il est important de bien faire la distinction entre une **méthode** qui est une démarche d'organisation et de conception en vue de résoudre un problème informatique, et le **formalisme** dont elle peut user pour exprimer le résultat.

Les grandes entreprises ont souvent leurs propres méthodes de conception ou de réalisation de projets informatiques. Celles-ci sont liées à des raisons historiques, d'organisation administrative interne ou encore à d'autres contraintes d'environnement (défense nationale, ...) et il n'est pas facile d'en changer. Il n'était donc pas réaliste de tenter de standardiser une méthodologie de conception au niveau mondial.

UML n'est pas une méthode, mais un langage. Il peut donc être utilisé sans remettre en cause les procédés habituels de conception de l'entreprise et, en particulier, les méthodes plus anciennes telles que celle proposée par OMT sont tout à fait utilisables. D'ailleurs, la société RATIONAL (principale actrice de UML) propose son propre processus de conception appelé OBJECTORY et entièrement basé sur UML. Ainsi, UML facilite la communication entre clients et concepteurs, ainsi qu'entre équipes de concepteurs. (diagramme UML), cela accélère

étant formellement définie dans [JBR97b] (sous forme de diagramme UML), cela accélère le développement des outils graphiques d'atelier de génie logiciel permettant ainsi d'aller de la spécification (haut niveau) en UML vers la génération de code (JAVA, C++, ADA, ...). De plus, cela autorise l'échange électronique de documents qui deviennent des spécifications exécutables en UML.

UML ne se contente pas d'homogénéiser des formalismes existants, mais apporte également un certain nombre de nouveautés telles que la modélisation d'architectures distribuées ou la modélisation d'applications temps réel avec gestion du multi-tâches, dont l'exposé dépasse le cadre de ce document.

II.5. Les objectifs d'UML :

- Représenter des systèmes entiers
 - Etablir un couplage explicite entre les concepts et les artefacts exécutables
 - Prendre en compte les facteurs d'échelle
 - Créer un langage de modélisation utilisable à la fois par les humains et les machines
- Recherche d'un langage commun :
- Utilisable par toutes les méthodes
 - Adapté à toutes les phases du développement
 - Compatible avec toutes les techniques de réalisation.

II.6. UML un langage

- UML n'est pas une méthode
- UML est un langage de modélisation objet
- UML a été adopté par toutes les méthodes objet
- UML est dans le domaine public, c'est une norme.

II.7. UML un langage pour :

➤ Visualiser

Chaque symbole graphique a une sémantique

➤ Spécifier

De manière précise et complète, sans ambiguïté,

➤ Construire

Les classes, les relations SQL peuvent être générées automatiquement

➤ Documenter

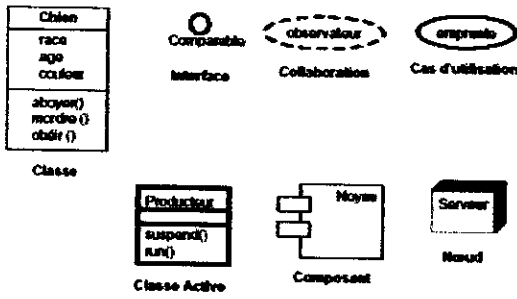
Les différents diagrammes, notes, contraintes, exigences seront présentés dans un document.

II.8. les domaines d'utilisation d'UML:

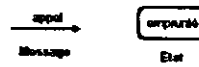
- Systèmes d'information des entreprises
- Les Banques et les services financiers
- Télécommunications
- Transport
- Défense et aérospatiale
- Scientifique
- Applications distribuées par le WEB

II.9. Les éléments de base en UML:

Les entités structurelles



Les entités de comportement

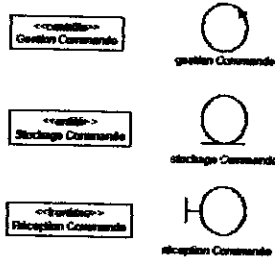


Les entités de groupement

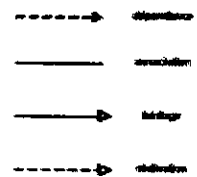


Les entités de notation

Séréotypes et icônes associées



Les relations



II.10. Les différents diagrammes d'UML:

Les 9 diagrammes en UML

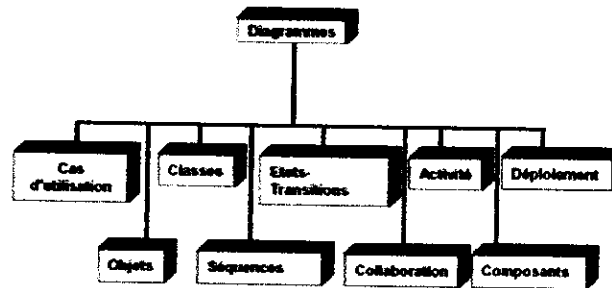


Figure. III. 2 – les diagrammes UML

II.11. Les façons de voir un système :

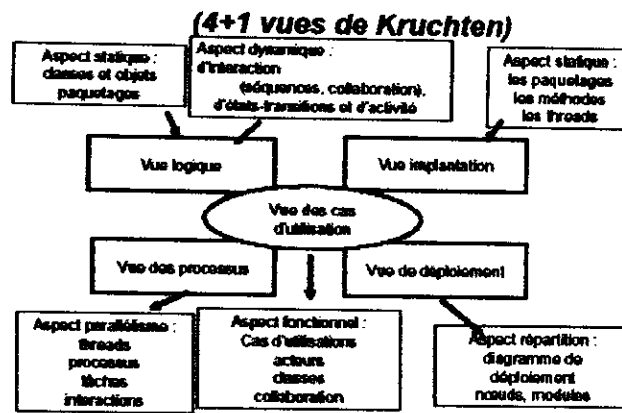


Figure.III.3 – vues de kruchten

II.12. Description des 9 diagrammes :

1. Diagramme de Classes:

- ◆ Un diagramme de classes est un graphe d'éléments connectés par des relations.
- ◆ Un diagramme de classes est une vue graphique de la structure statique d'un système

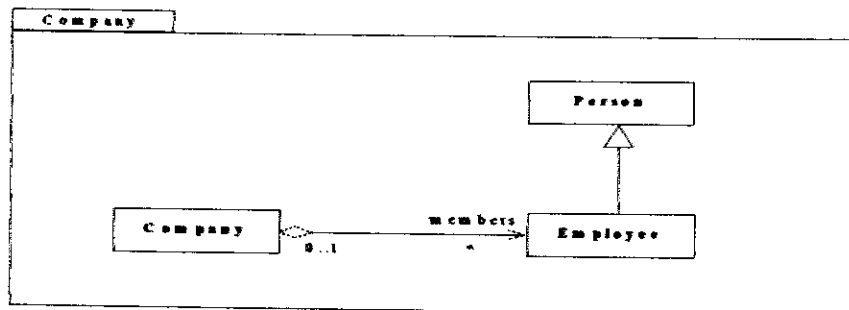
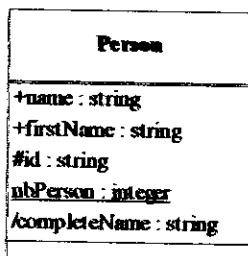


Figure.III.4. – Exemple de diagramme de classe

- Une classe représente la structure commune d'un ensemble d'objets.
- Une classe est représentée par un rectangle qui contient une chaîne de caractères correspondant au nom de la classe
- Ce rectangle peut être séparé en trois parties (nom, attributs, opérations).
- Le nom de la classe doit commencer par un caractère alphabétique et ne pas contenir le caractère ':'

Classes:

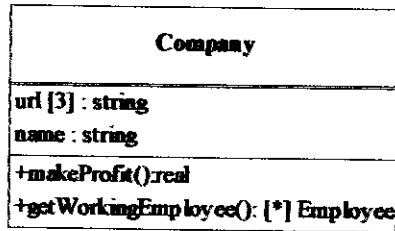


Attributs:

- Une classe peut contenir des attributs
- La syntaxe d'un attribut est : *visibilité nom : type*
- La visibilité est:
 - '+' pour public
 - '#' Pour protected
 - '-' pour private
- UML définit son propre ensemble de types
Integer, real, string, ...
- Un attribut peut être un attribut de classe, il est alors souligné.
- Un attribut peut être dérivé, il est alors préfixé par le caractère '/'

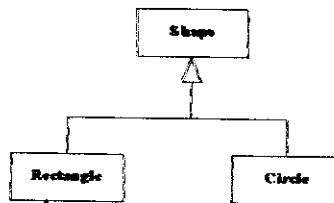
Opérations:

- Une opération est un service qu'une instance de la classe peut exécuter
- La syntaxe d'une opération est: *visibility name(parameter):return*
- La syntaxe des paramètres est: *kind name : type*
- Le kind peut être: in, out, inout



Héritage:

- L'héritage est une relation entre un élément plus général et un élément plus spécifique. L'héritage existe entre des classes, des packages, ...
- L'héritage multiple est possible en UML

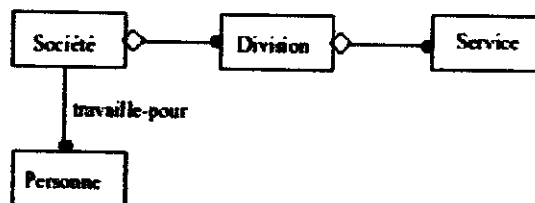


Associations:

- "1" signifie un exactement
 - "1..*" signifie de un à plusieurs
 - "*" et "0..*" signifient de zéro à plusieurs
 - "3..5" signifie l'intervalle 3 à 5 inclus
 - "2, 4, 18" signifie explicitement les valeurs 2, 4, 18
 - * = plusieurs
- par défaut, une ligne simple = un

Agrégation & association:

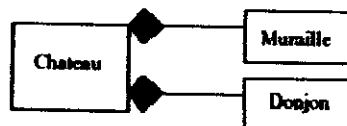
L'agrégation est un cas particulier d'association.



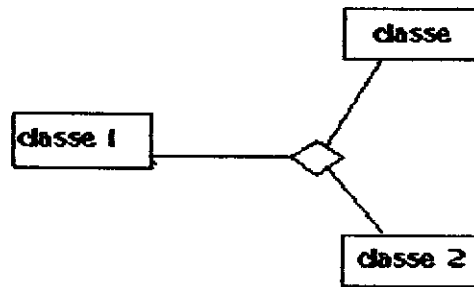
Agrégation et association

Composition & Agrégation :

UML propose la notion de composition. Cette notion est très proche de celle d'agrégation. La Composition est une agrégation *réalisée par valeur*. Elle se note avec un rectangle noir.



- Les associations N-aires connectent plusieurs éléments entre eux.
- Les associations N-aires sont très peu utilisées.



Classes-Associations :

- Une classe-association est une association qui est aussi une classe.
- Les classes-associations sont utilisées lorsque les associations doivent porter des informations
- Il est toujours possible de se passer des classes-associations.

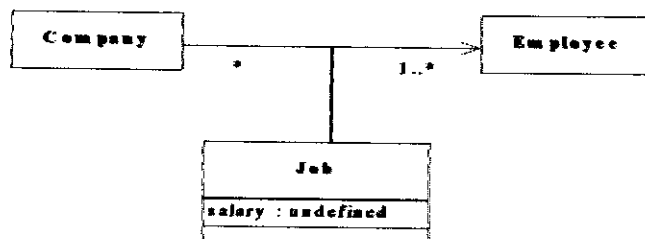
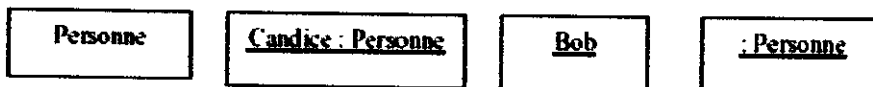


Figure .III. 5 – Exemple de calsses _association

2. Diagramme d’Objets:

Les diagrammes d’objets permettent de modéliser les instances. En UML, le mot « objet » est souvent lié à la notion d’instance alors qu’en orienté objet usuel, le mot « objet » est souvent lié aux deux notions de classe et d’instance.

Exemple : une classe et une instance (objet) :



« Personne » est une classe. « Candice » est une instance qui appartient à la classe« Personne ». « Bob » est une instance dont la classe n’est pas précisée. La troisième instance est un objet anonyme dont la classe est précisée mais pas le nom.

■ Un diagramme d’objet représente la vue statique d’un ensemble d’instance de classes

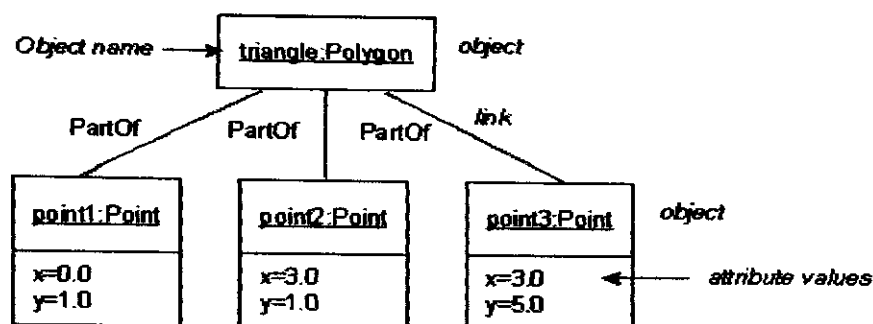


Figure .III. 6 – Exemple de diagramme d’objet

3. Diagramme de Collaboration :

Un diagramme de collaboration représente la vue statique et la vue dynamique d'un ensemble d'éléments

Une collaboration définit des rôles (et non pas des classes !).

Un diagramme de flux ou *diagramme de collaboration* est un diagramme sur lequel les classes d'objets sont représentées avec des rectangles reliés par des flèches représentant les flux d'événements entre les classes.

Exemple :

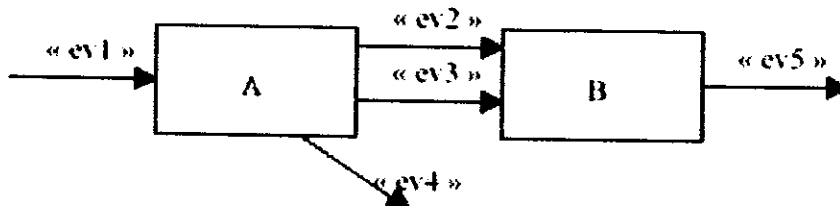


Figure. .III. 7 – Exemple de diagramme de collaboration

Ici, l'événement « ev1 » est recevable par la classe 'A'. Les événements « ev2 », « ev3 » et « ev4 » sont émissibles par 'A'. Les événements « ev2 » et « ev3 » sont recevables par la classe

'B'. Enfin, l'événement « ev5 » est émissible par la classe 'B'.

Ce type de diagramme sert à répertorier tous les événements, reçus et envoyés, relatifs à chaque classe.

4. Diagramme de séquence:

Pour commencer à décrire l'évolution d'un ensemble d'objets, il est possible de dessiner d'abord un *diagramme de séquence*.

Horizontalement, on place les instances concernées par un scénario et on relie les instances par des flèches indiquant le flux d'événements. Verticalement, le temps est représenté. Ce type de diagramme donne une première idée des événements qui pourront être pertinents dans la modélisation. On répète ce diagramme autant de fois qu'il existe de scénarii d'événements possibles

Exemple :

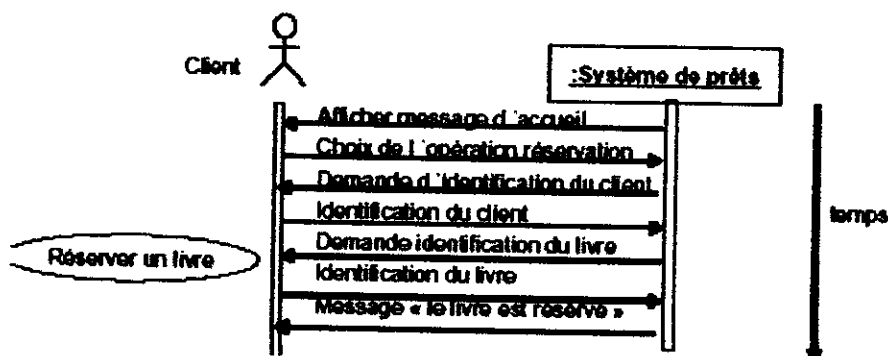
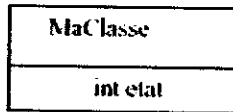


Figure. .III. 8 – Exemple de diagramme de séquence

5. Diagramme d'état transitions:

Etat :

Un objet possède un *état* à un instant donné. L'état de l'objet est une notion durable à l'échelle de temps d'évolution des objets. Par exemple, si l'objet est un dé à jouer, on peut dire que le dé possède 6 états possibles et que le dé posé sur la table est dans un état durable (tant qu'on ne le lance pas). L'état peut être spécifié explicitement :



L'état peut aussi ne pas être explicitement présent sous forme d'un attribut. Dans ce cas, l'état de l'objet au sens strict correspond à l'ensemble des attributs de l'objet. Et l'état au sens large correspond à l'ensemble des attributs et liens de l'objet.

- Un diagramme d'état représente la vue dynamique d'un ensemble d'éléments sous forme d'état

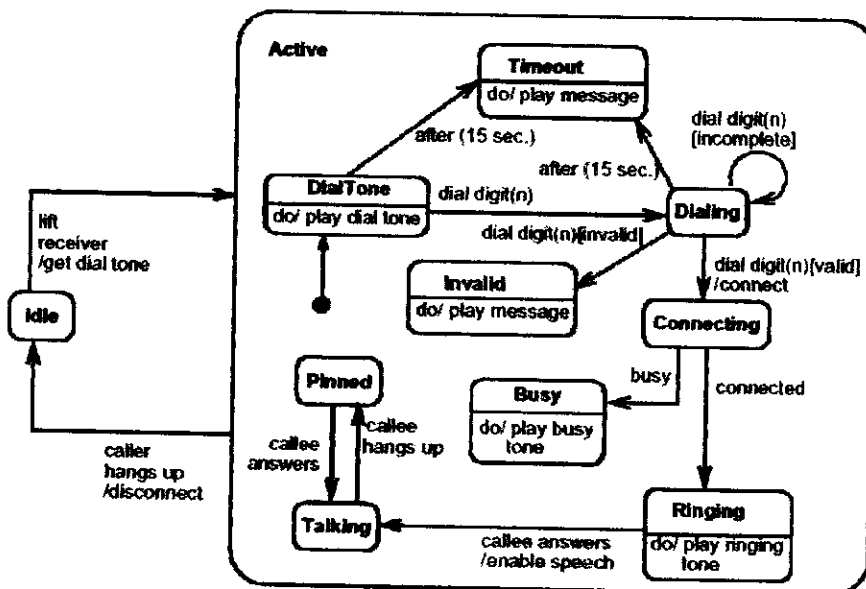


Figure. III. 9 – Exemple de diagramme d'état-transition

5. Diagrammes d'activités:

Un diagramme d'activité représente la vue dynamique d'un ensemble d'éléments sous de flux d'exécution

Le diagramme d'activité est un cas particulier de diagramme d'états, dans lequel à chaque état correspond une activité constituant un élément d'une tâche globale à réaliser.

Le but de ce diagramme est de mettre en évidence les contraintes de séquentialité et de parallélisme qui pèsent sur la tâche globale.

Ainsi, dans l'exemple suivant, on voit que, pour se faire un café, on peut simultanément mettre un filtre à la cafetière, remplir le réservoir d'eau et prendre une tasse mais que, par contre, il faut attendre d'avoir mis un filtre pour mettre du café.

Exemple:

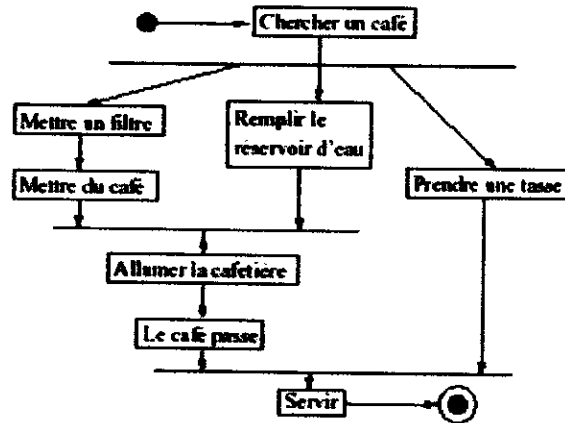


Figure. III. 10 – Exemple de diagramme d'activité

6. Le modèle des cas d'utilisation :

Les cas d'utilisation (use cases) ont été formalisés par Ivar Jacobson. Ils décrivent sous forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur. Avant UML, ils n'étaient pas formalisés par les autres méthodes objet telles que OMT.

Les cas d'utilisation sont utiles lors de l'élaboration du cahier des charges ou du document de spécifications des besoins du logiciel.

Le modèle des cas d'utilisation comprend les acteurs, le système et les cas d'utilisation.

L'ensemble des fonctionnalités du système est déterminé en examinant les besoins de chaque acteur, exprimés sous forme de famille d'interactions dans les cas d'utilisation. Les acteurs se représentent sous forme de petits personnages qui déclenchent les cas. Ces derniers se représentent par des ellipses contenues dans un rectangle représentant le système.

Exemple:

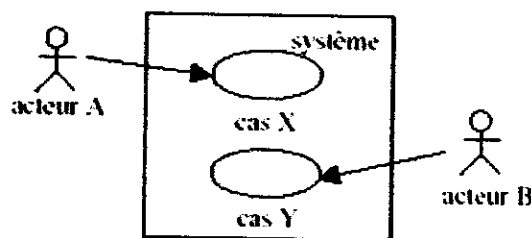


Figure. III. 11 – Exemple de diagramme de cas d'utilisation

Dans cet exemple, l'acteur A déclenche le cas X et l'acteur B déclenche le cas Y. Il existe quatre catégories d'acteurs :

- les acteurs principaux,
- les acteurs secondaires,
- le matériel externe,
- les autres systèmes.

Chaque acteur doit être décrit en 3 ou 4 lignes de manière claire et concise.

Un cas d'utilisation décrit un ensemble de scénarios du point de vue de l'utilisateur grâce à des diagrammes de séquence ou des diagrammes de collaboration.

Liens entre cas d'utilisation : include et extend

Il est parfois intéressant d'utiliser des liens entre cas (sans passer par un acteur), UML en fournit de deux types : la relation **utilise** (include) et la relation **étend** (extend).

Utilisation de cas : La relation **utilise** (include) est employée quand deux cas d'utilisation ont en commun une même fonctionnalité et que l'on souhaite factoriser celle-ci en créant un sous-cas, ou cas intermédiaire, afin de marquer les différences d'utilisation.

Extension de cas (extend) : Schématiquement, nous dirons qu'il y a extension d'un cas d'utilisation quand un cas est globalement similaire à un autre, tout en effectuant un peu plus de travail (voire un travail plus spécifique). Cette notion – à utiliser avec discernement – permet d'identifier des cas particuliers (comme des procédures à suivre en cas d'incident) dès le début ou lorsque l'attitude face à un utilisateur spécifique du système doit être spécialisée ou adaptée. Il s'agit grosso modo d'une variation du cas d'utilisation normale.

Exemple:

Par exemple, dans le cas d'un distributeur automatique de billets dans une banque, les utilisateurs du distributeur qui sont clients de la banque peuvent effectuer des opérations qui ne sont pas accessibles à l'utilisateur normal (par exemple, consultation de solde).

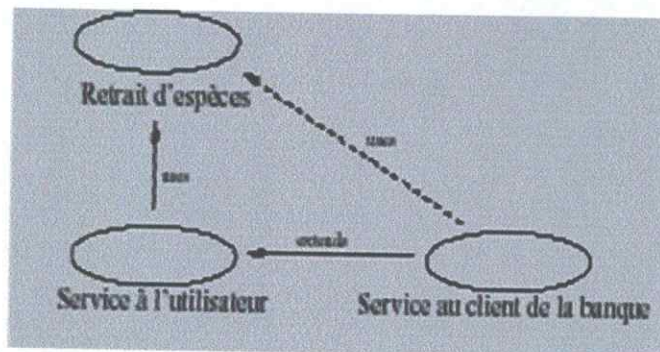


Figure. .III. 12 – Exemple d'extension et d'utilisation

8. Diagramme de composant :

Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en terme de modules : fichiers sources, bibliothèques, exécutables, etc. Ils montrent la mise en oeuvre physique des modèles de la vue logique avec l'environnement de développement.

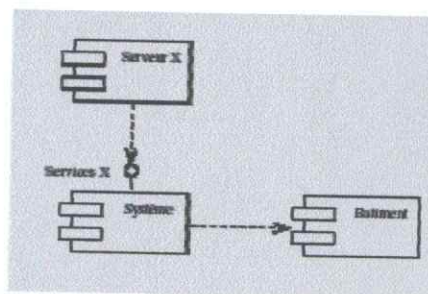


Figure. .III.13. Exemple diagramme de composants

9. Diagramme de déploiement :

Illustre la disposition physique des différents matériels (ou nœuds) qui entrent dans la composition du système.

- Illustre la répartition des composants (cf. diagrammes de composants) au sein des nœuds.
- Les diagrammes de déploiement existent sous deux formes : spécification et instance.
- Un diagramme est composé :
 - de dispositifs physiques (les **nœuds**),
 - d'objets d'implantation attachés aux nœuds (les **composants**),

– de liens représentant les moyens de communication entre les nœuds (les **supports de communication**).

- Chaque ressource matérielle est représentée par un nœud.
- En général, cette ressource possède ses propres attributs (Capacité mémoire, capacité calculatoire, ...).

Exemple :

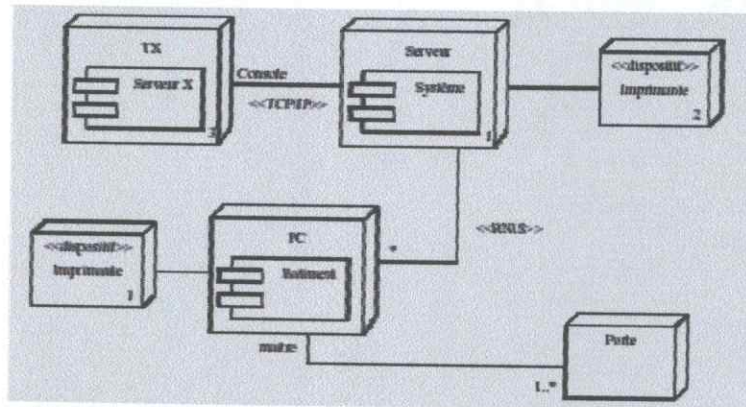


Figure. .III.14. Exemple diagramme de déploiement

III. La conception du nouveau système en utilisant UML :

III.1. modèle fonctionnel :

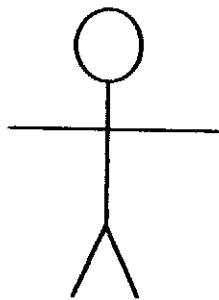
1. les cas d'utilisations :

1.1 les acteurs :

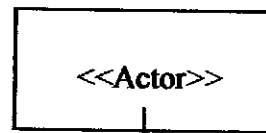
➤ définition générale :

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

Les acteurs sont représentés soit sous forme graphique dite : **stick man**, soit sous forme rectangulaire avec le mot clé : **<<Actor>>**.



Stick man



le mot clé



➤ les acteurs existants :

L'administrateur : il gère le système (ajout de nouvelles agences, nouveaux producteurs, attribution de droits et de privilèges aux producteurs, configuration de nouveaux mots de passe).

L'utilisateur : sont les producteurs des agences, ils manipulent les différentes ressources de système (mise à jour, consultations).

1.2. La liste des cas d'utilisation :

Légende :

Adm : administrateur

Utl : utilisateur

Cas d'utilisation	Acteurs
Authentification	Adm/utl
Configurer mot de passe	adm
créer agence	adm
Supprimer agence	adm
créer délégation	adm
Supprimer délégation	adm
Consulter agence	adm
Consulter délégation	adm
production	Adm/utl
sinistre	utl
Créer police	Utl
Créer avenant	Utl
Créer sinistre	Utl
Créer adhérent	Utl
Créer remboursement	Utl
Supprimer police	Utl
Supprimer adhérent	Utl
Consulter police	Utl/adm
Consulter avenant	Utl/adm
Consulter remboursement	Utl/adm
Consulter adhérent	Utl/adm
Consulter agence	Adm
Consulter délégation	Adm
Consulter sinistres	Utl/adm

Tableau III.1 : listes des cas d'utilisation

1.3 Les diagrammes des cas d'utilisations :

➤ Authentification :

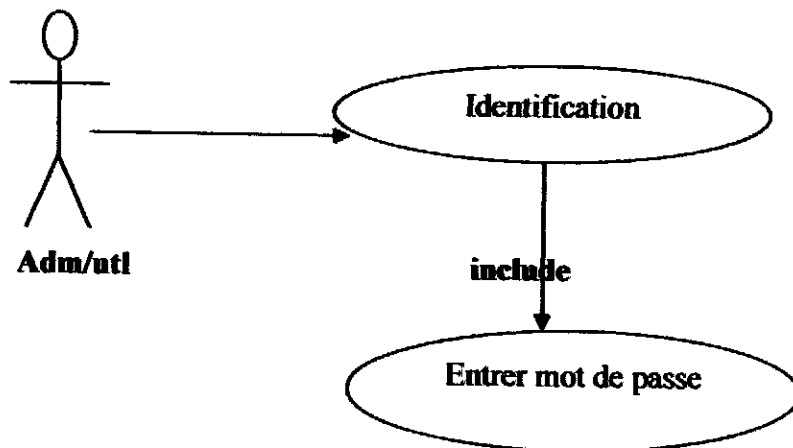


Figure. III.15.diagramme cas d'utilisation « Authentification »

➤ Configurer mot de passe :

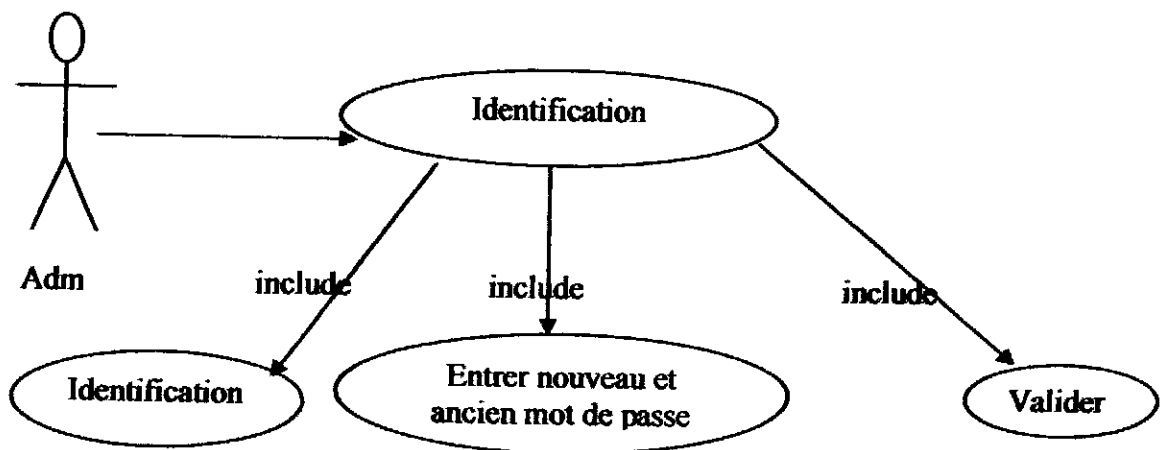


Figure. III.16.diagramme cas d'utilisation « Configurer mot de passe »

➤ **Créer agences/délégation :**

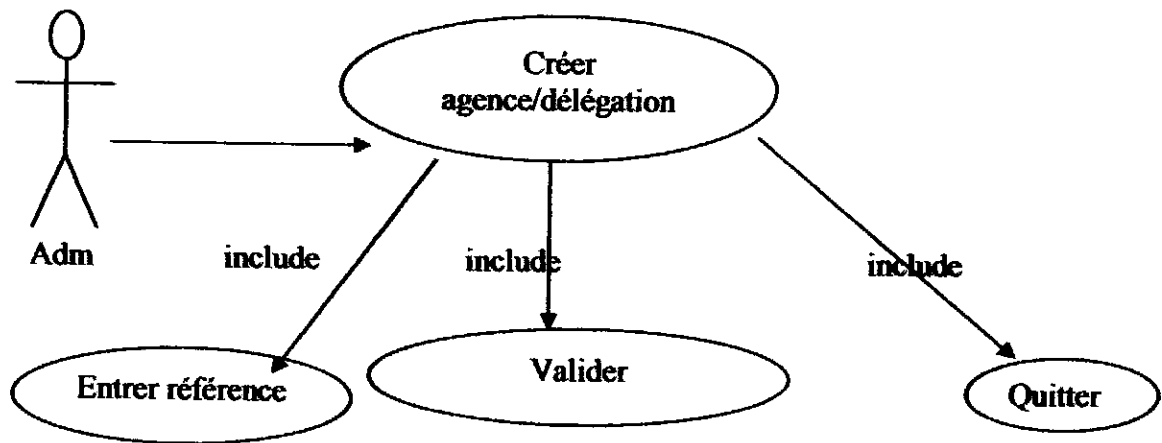


Figure. .III.17.diagramme cas d'utilisation « Créer agence,délégation »

➤ **Supprimer agence / délégation :**

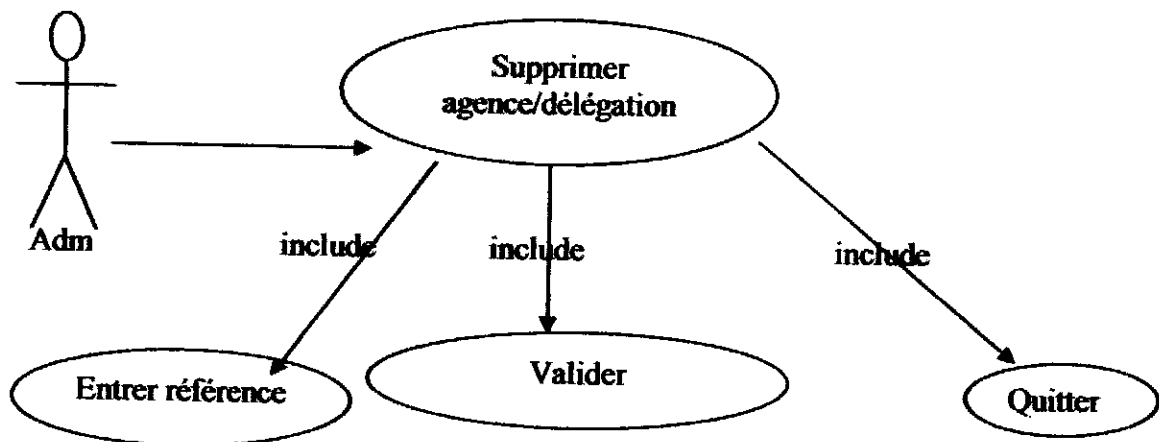


Figure. .III.18.diagramme cas d'utilisation « supprimer agence,délégation »

➤ Créer police :

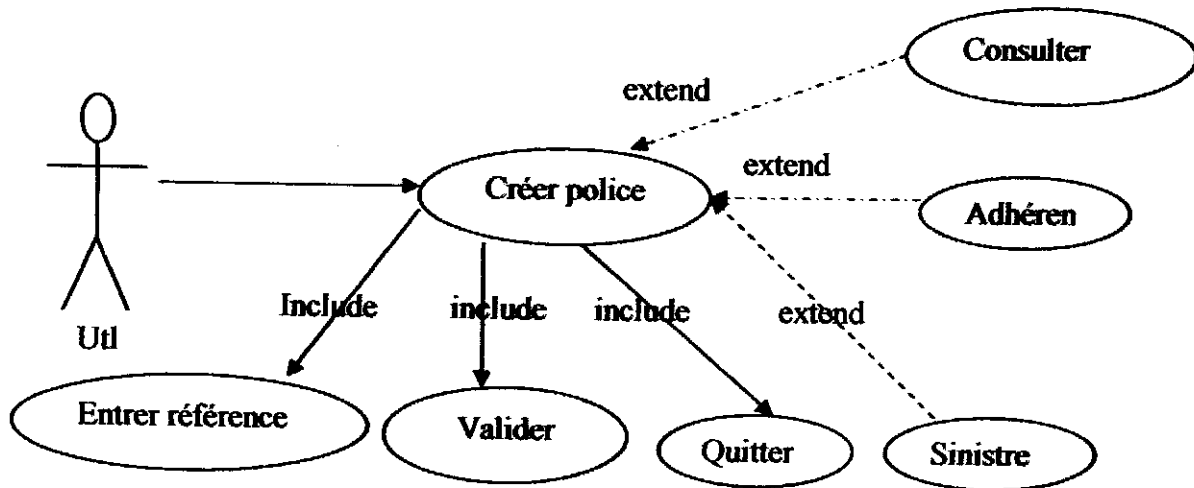


Figure .III.19.diagramme cas d'utilisation « Créer police »

➤ Créer avenant :

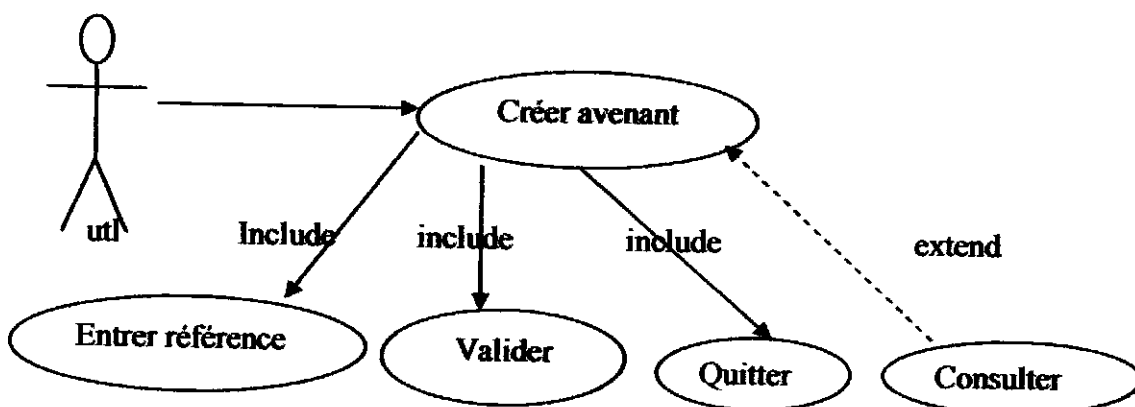


Figure .III.20.diagramme cas d'utilisation « Créer avenant »

➤ Créer sinistre :

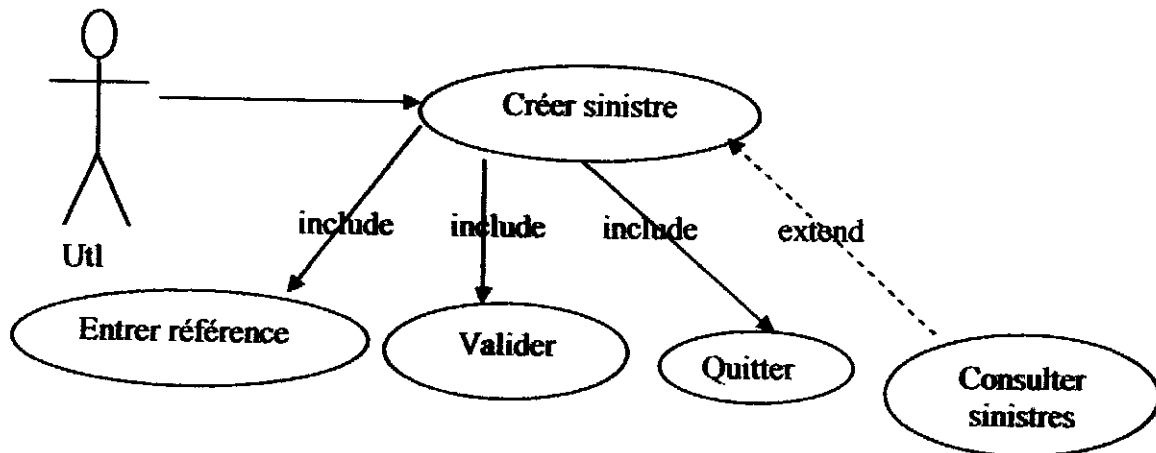


Figure .III.21.diagramme cas d'utilisation « Créer sinistre »

➤ Créer remboursement :

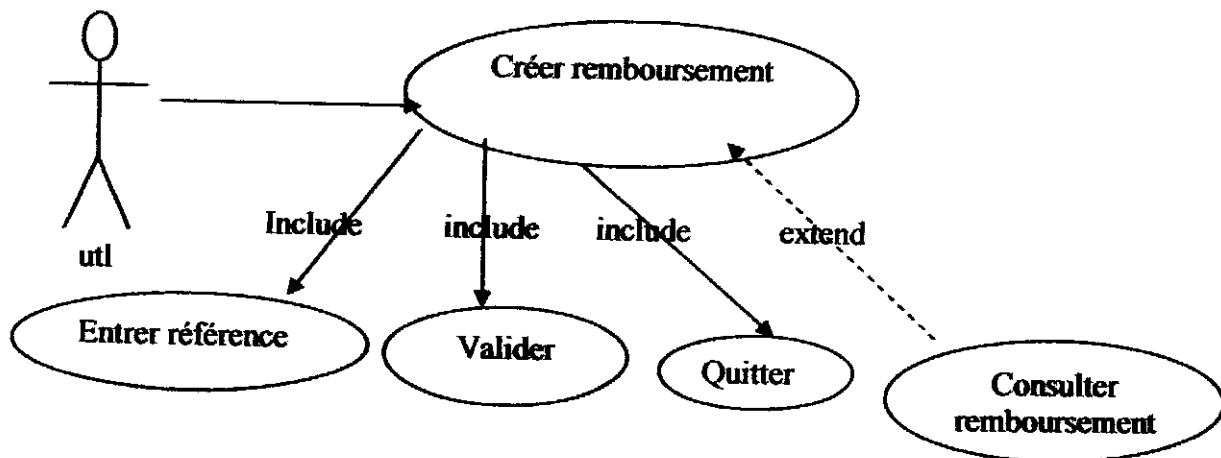


Figure .III.22.diagramme cas d'utilisation « Créer remboursement »

➤ Consulter police :

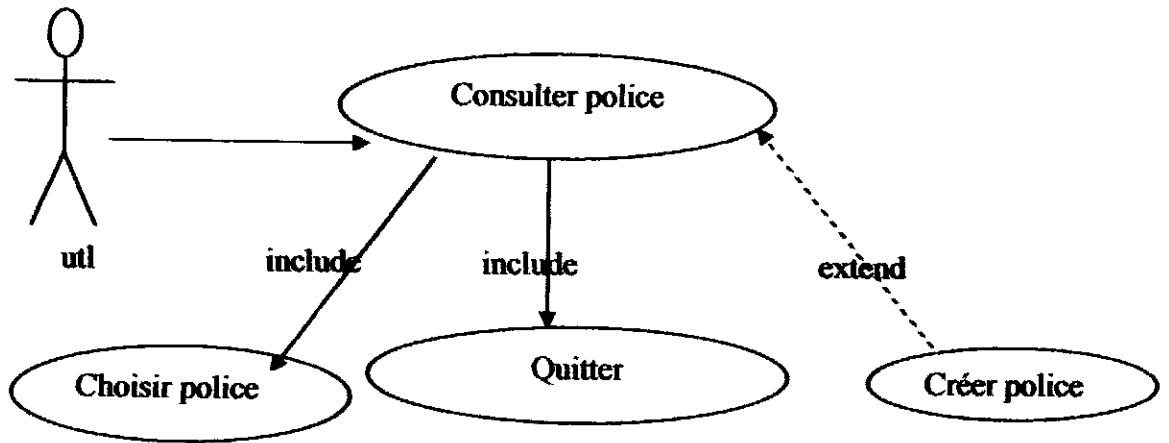


Figure. .III.23.diagramme cas d'utilisation « consulter police »

➤ Consulter avenant :

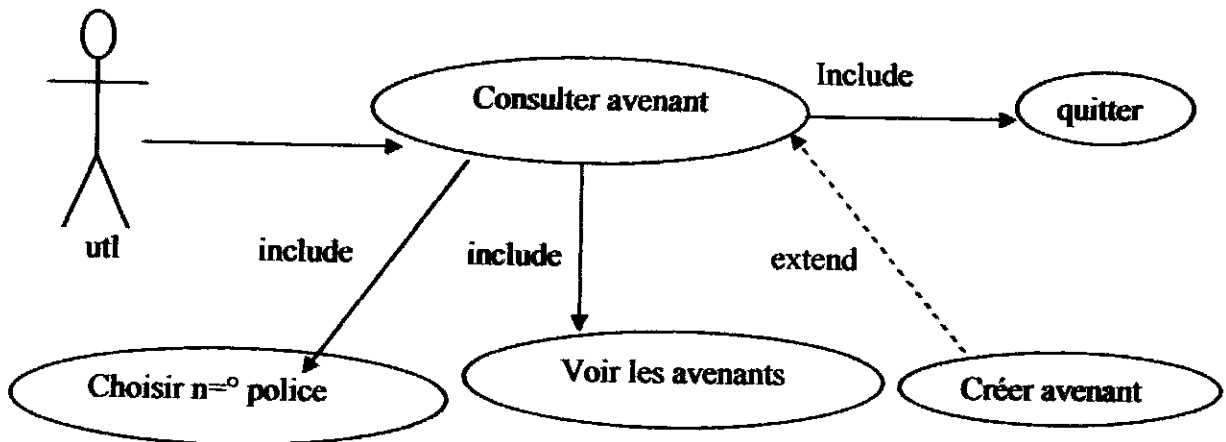


Figure. .III.24 diagramme cas d'utilisation « consulter Avenant »

➤ Consulter remboursement :

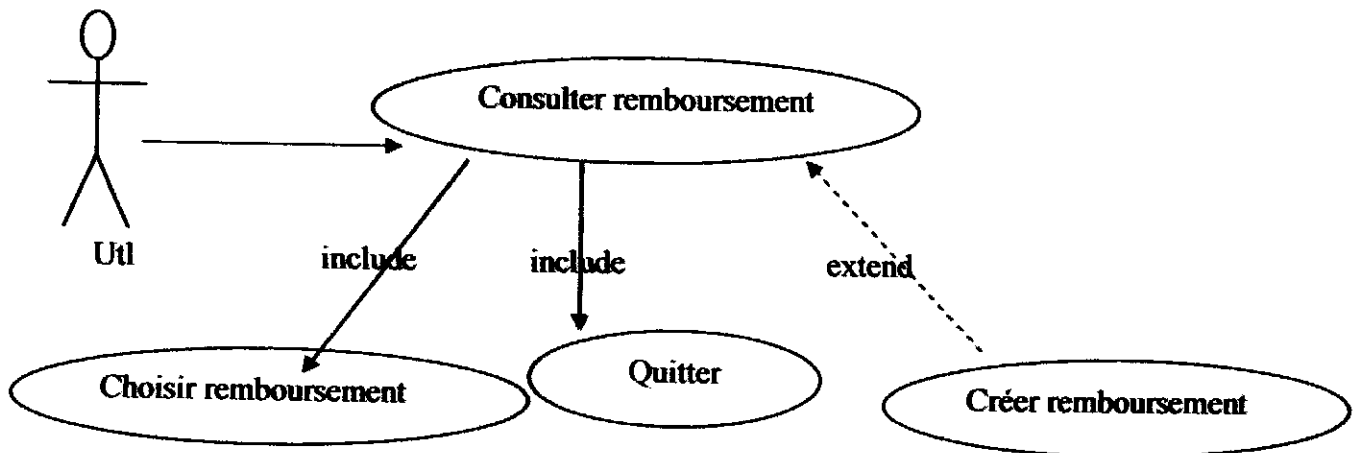


Figure .III.25.diagramme cas d'utilisation « consulter remboursement »

➤ Consulter adhérent :

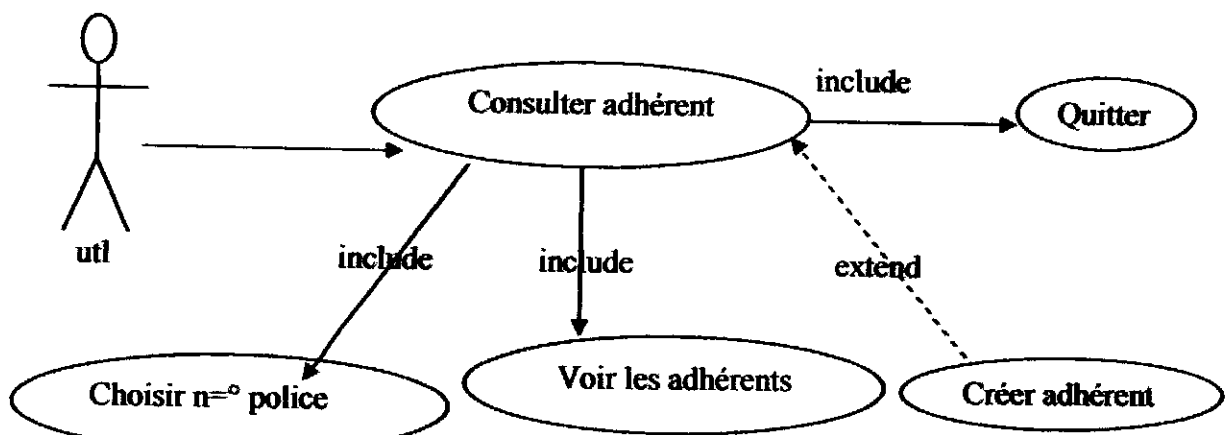


Figure .III.26 diagramme cas d'utilisation « consulter Adhérent »

➤ **Supprimer police/adhérent :**

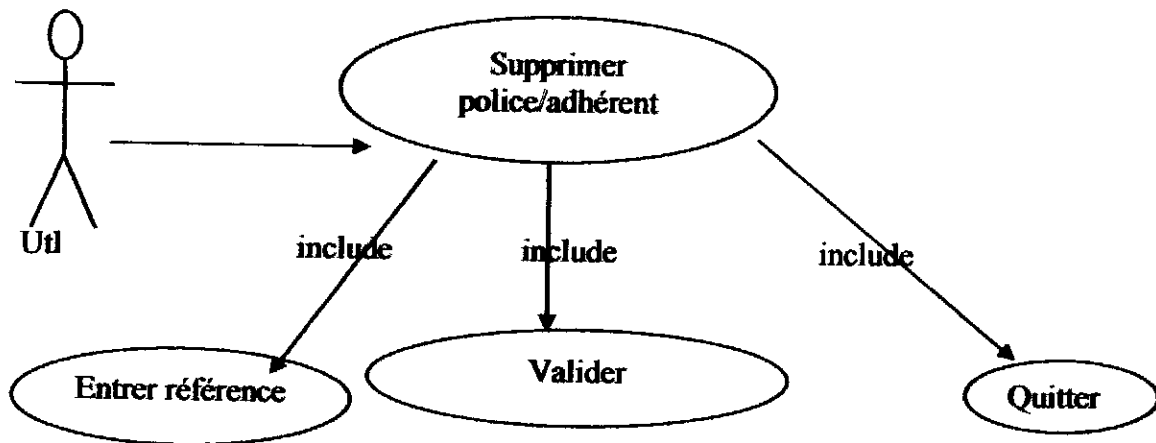


Figure .III.27 diagramme cas d'utilisation « supprimer police/adhérent »

➤ **Consulter agences:**

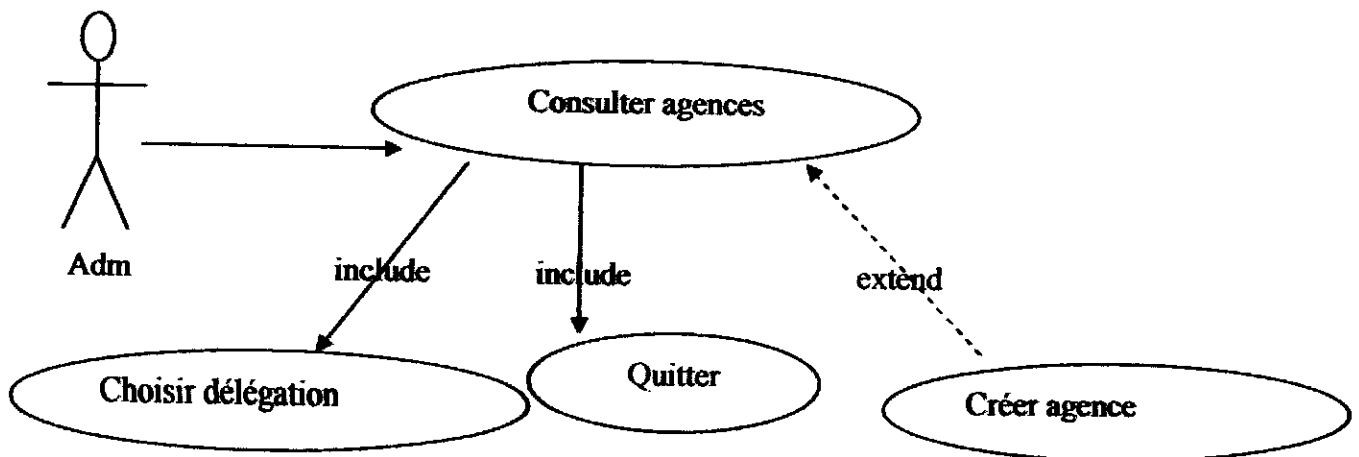


Figure.III.28 diagramme cas d'utilisation « consulter agence »

➤ Consulter délégation :

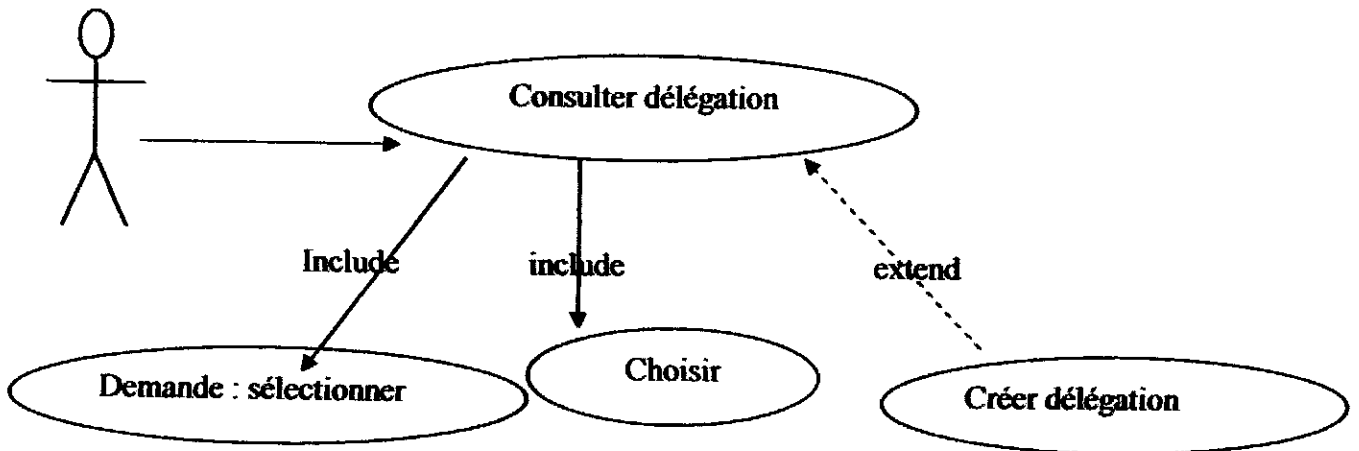


Figure.III.29 diagramme cas d'utilisation « consulter délégation »

➤ diagramme des cas d'utilisation globale :

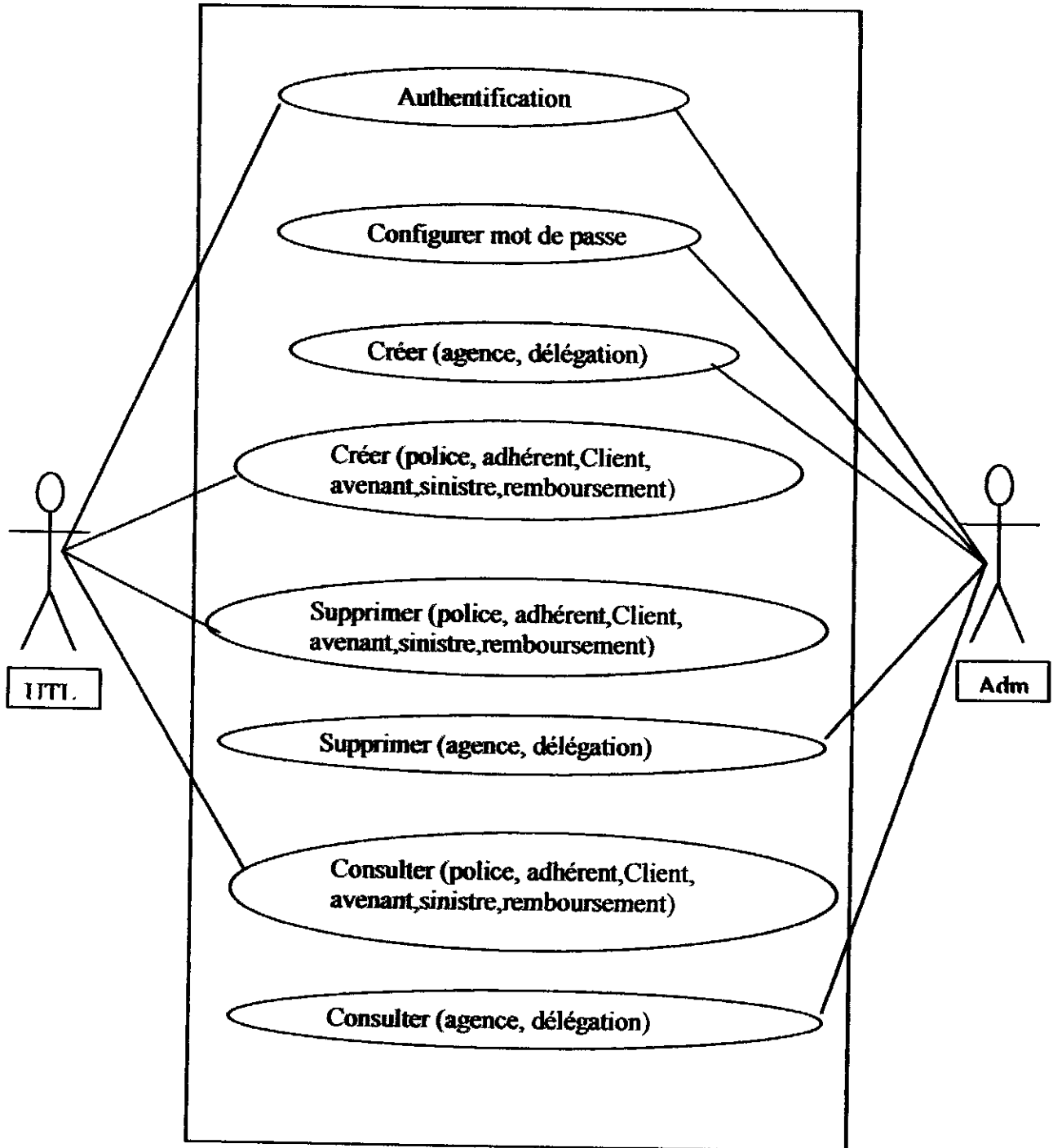


Figure. .III.30 diagramme des cas d'utilisation globale

2. les diagrammes de séquence :

➤ identification avec succès:

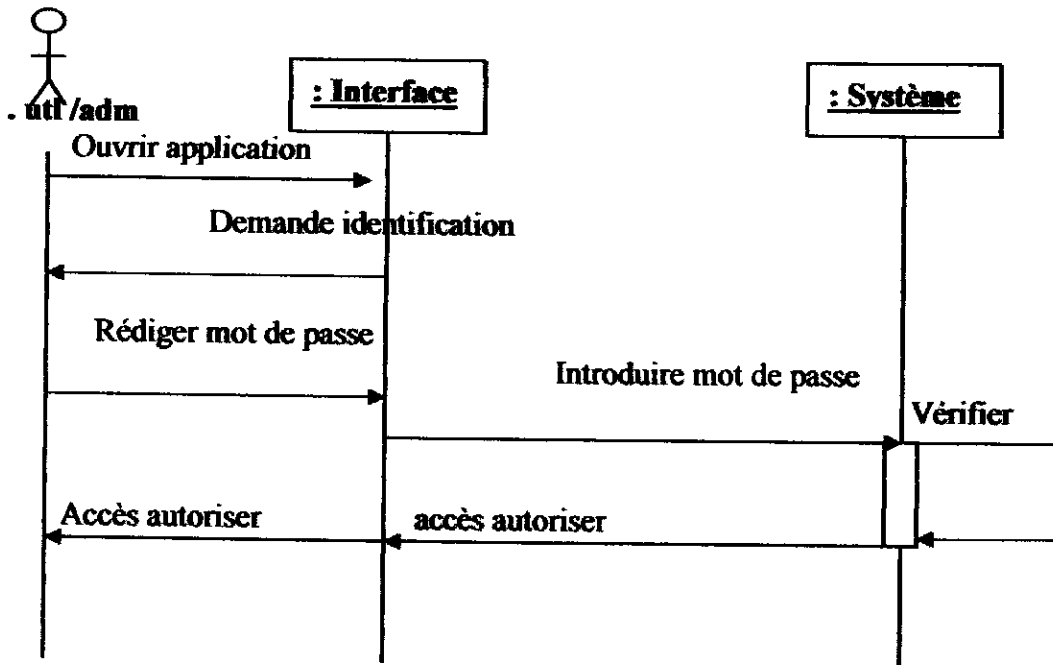


Figure.III.31 diagramme de séquence « identification par succès »

Scénario :

Pour pouvoir accéder au système (utilisateur ou administrateur), il faut s'identifier (introduire mot de passe), le système vérifie le mot de passe, pas d'erreur, accès autorisé.

➤ **identification avec erreur:**

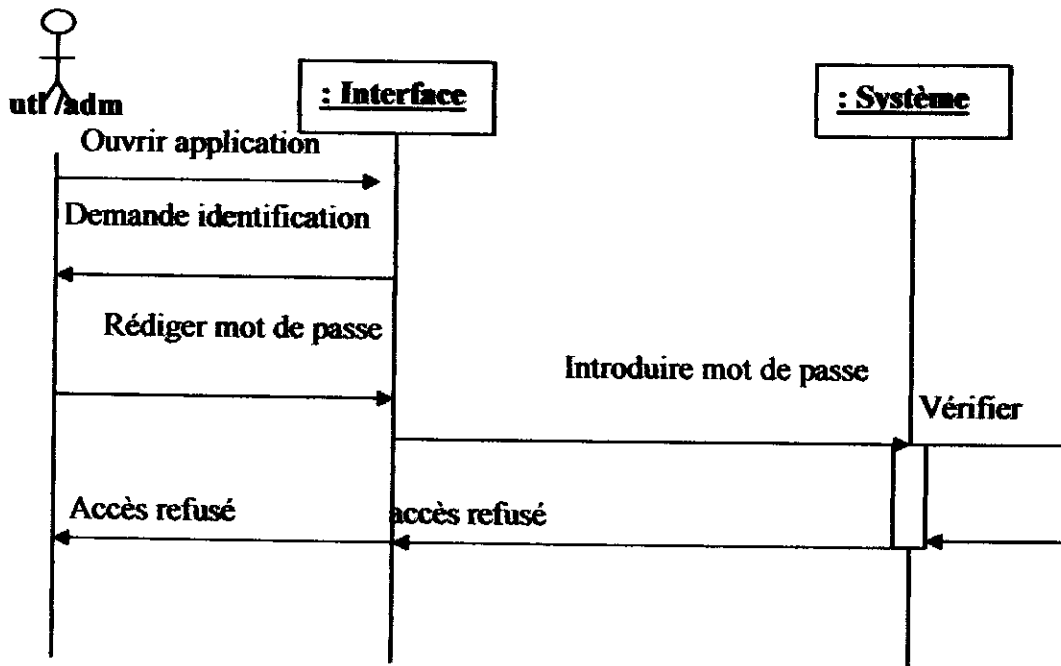


Figure .III.32 diagramme de séquence « identification avec erreur »

Scénario :

Pour pouvoir accéder au système (utilisateur ou administrateur), il faut s'identifier (introduire mot de passe), le système vérifie le mot de passe, erreur, accès refusé.

➤ **Ajouter police :**

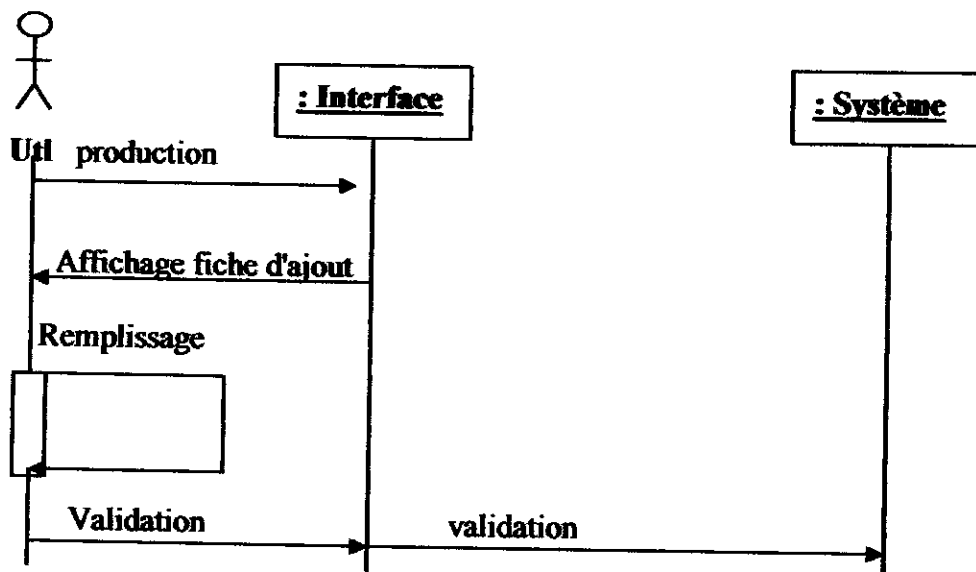


Figure .III.33 diagramme de séquence « ajouter police »

Scénario :

Le producteur d'agence demande la fiche d'ajout à partir la demande : production, la fiche d'ajout apparaît, le producteur remplit la fiche puis valide l'ajout.

➤ **Ajouter avenant :**

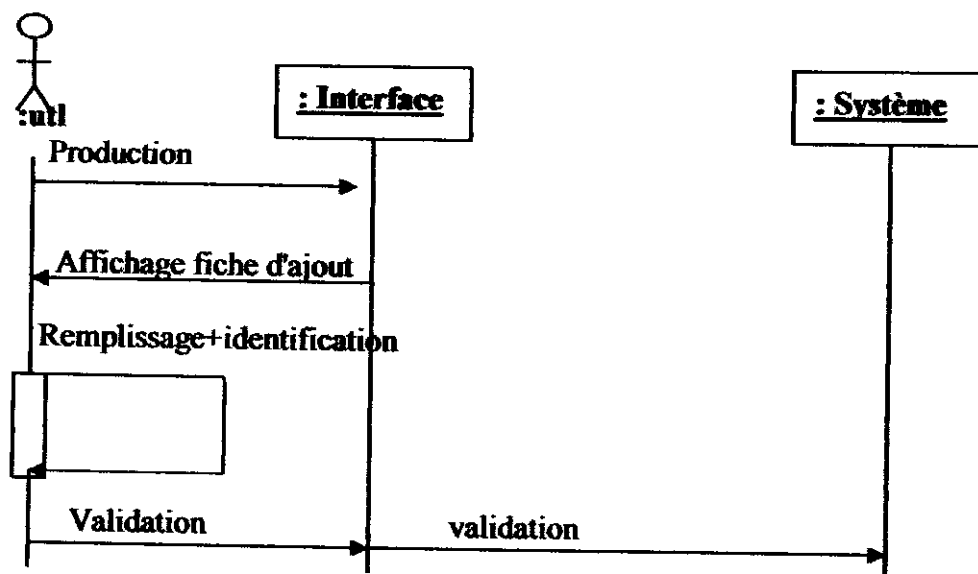


Figure .III.34.diagramme de séquence« ajouter avenant»

Scénario :

Le producteur d'agence demande la fiche d'ajout à partir la demande : production, la fiche d'ajout apparaît, le producteur remplit la fiche avec la sélection d'un numéro de police pour le faire un avenant puis valide l'ajout.

➤ **Ajouter un sinistre :**

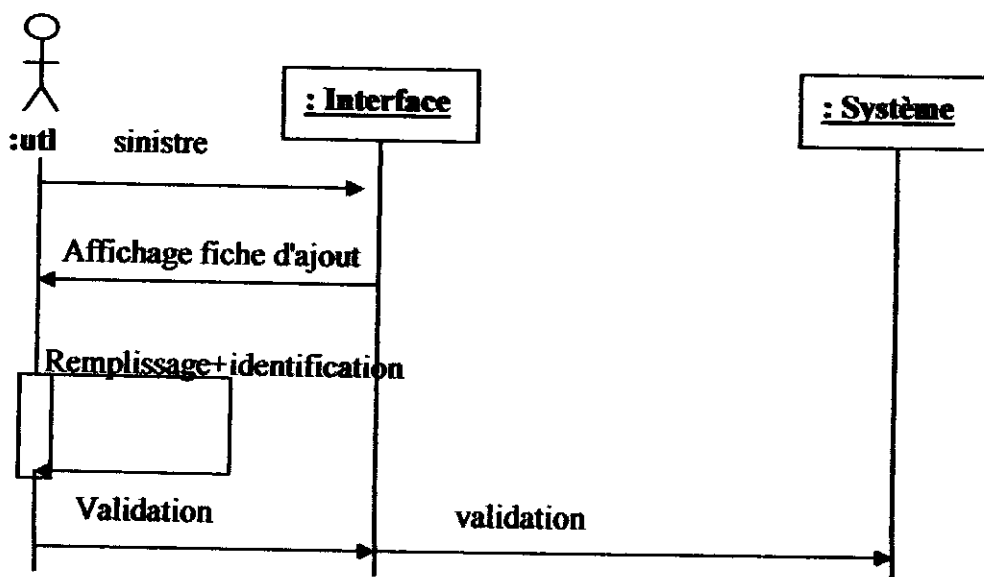


Figure .III.35 diagramme de séquence« ajouter sinistre»

Scénario :

Le producteur d'agence demande la fiche d'ajout à partir la demande : sinistre, la fiche d'ajout apparaît, le producteur remplit la fiche avec la sélection d'un numéro de police pour le faire une déclaration de sinistre puis valider l'ajout.

➤ **Ajouter remboursement :**

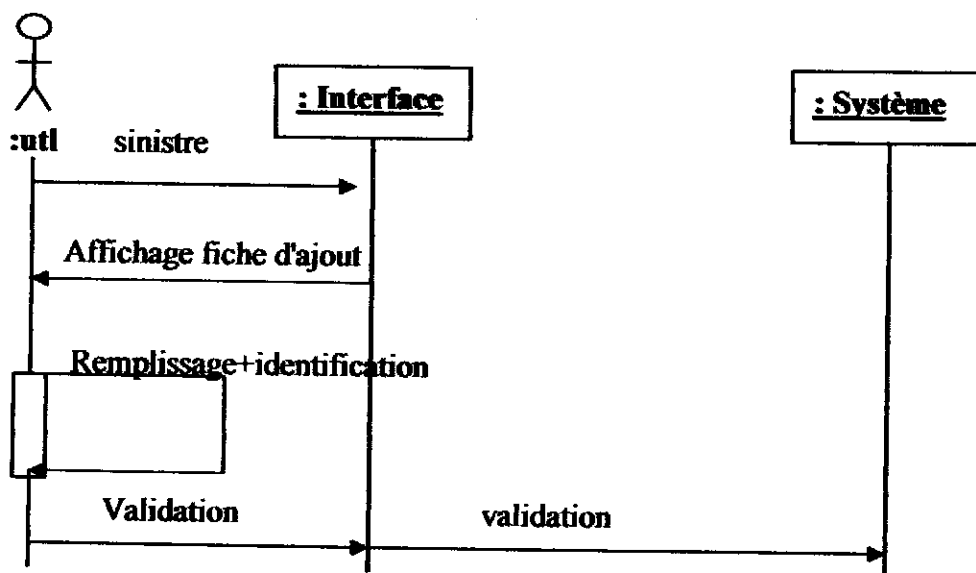


Figure. III.36.diagramme de séquence« ajouter remboursement»

Scénario :

Le producteur d'agence demande la fiche d'ajout à partir la demande : sinistre, la fiche d'ajout apparaît, le producteur remplit la fiche avec la sélection d'un numéro de police pour le déclarer q'un sinistre est réglé puis valider l'ajout.

➤ Ajouter agence/délégation :

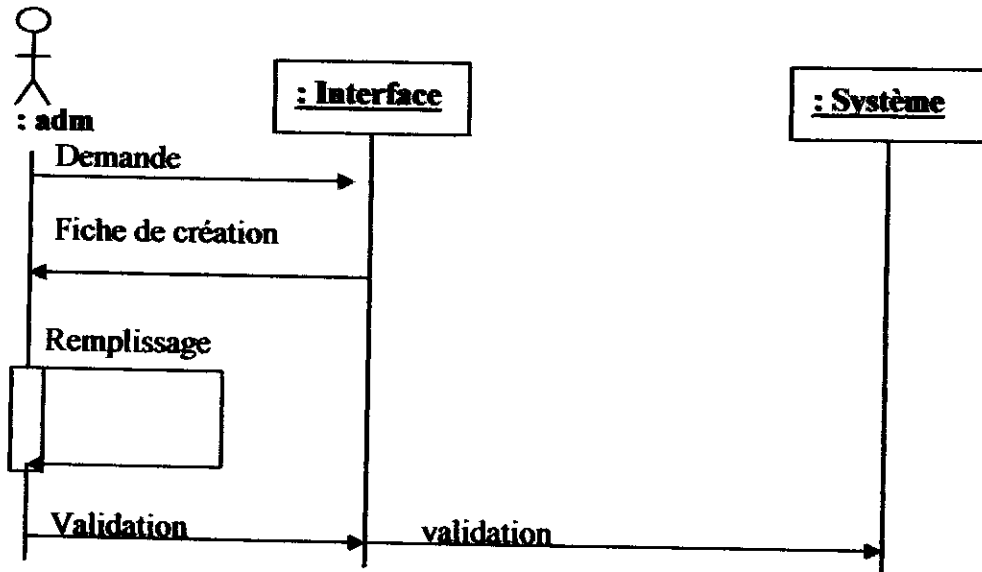


Figure. .III.37.diagramme de séquence« ajouter agence/délégation»

Scénario :

L'administrateur demande la création d'une nouvelle agence ou délégation, la fiche de création apparaît, l'administrateur introduit les cordonnés nécessaire puis valider la création.

➤ Configurer mot de passe:

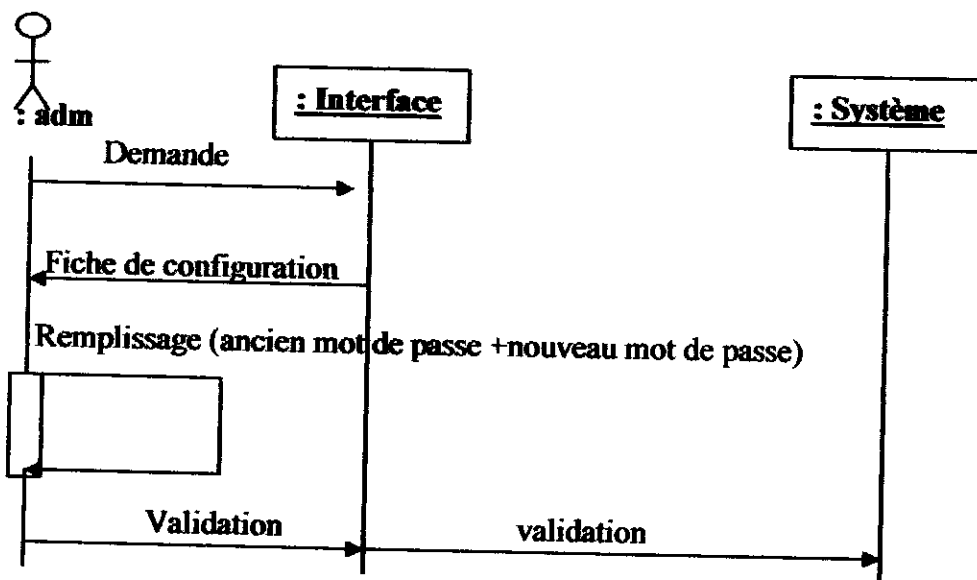


Figure. .III.38.diagramme de séquence« configurer mot de passe»

Scénario :

L'administrateur demande la configuration de mot de passe, la fiche de configuration apparaît, l'administrateur introduit l'ancien et nouveau mot de passe puis valider la configuration.

➤ **supprimer agence/délégation:**

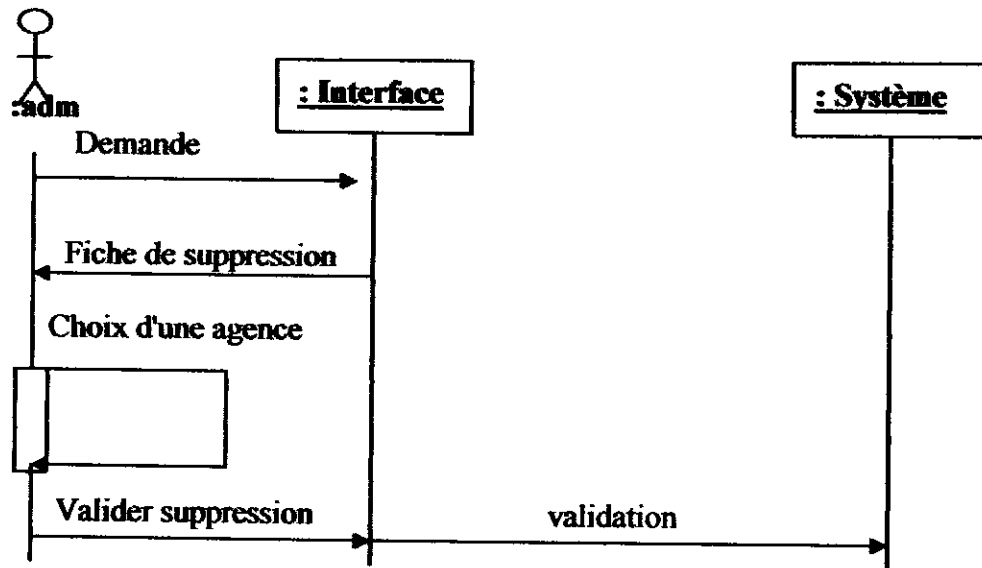


Figure. III.39.diagramme de séquence« supprimer agence/délégation»

Scénario :

L'administrateur demande la suppression d'une agence ou délégation, la fiche de suppression apparaît, l'administrateur choisit une agence/délégation à supprimer puis valide la suppression.

➤ **Supprimer police :**

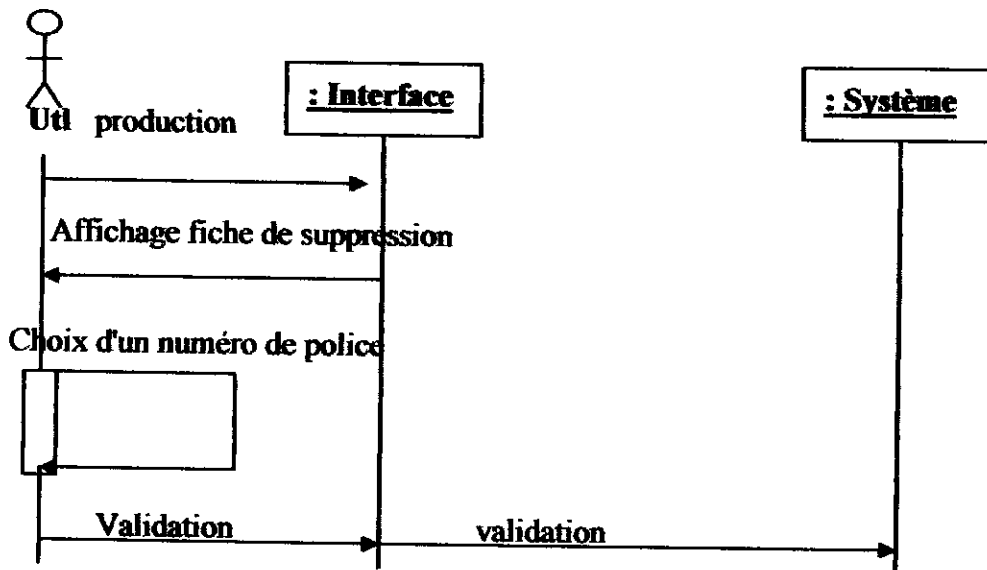


Figure. III.41diagramme de séquence« supprimer police»

Scénario :

Le producteur demande la suppression d'une police, la fiche de suppression apparaît, le producteur choisit un numéro de police à supprimer puis valide la suppression.

➤ **Consulter polices:**

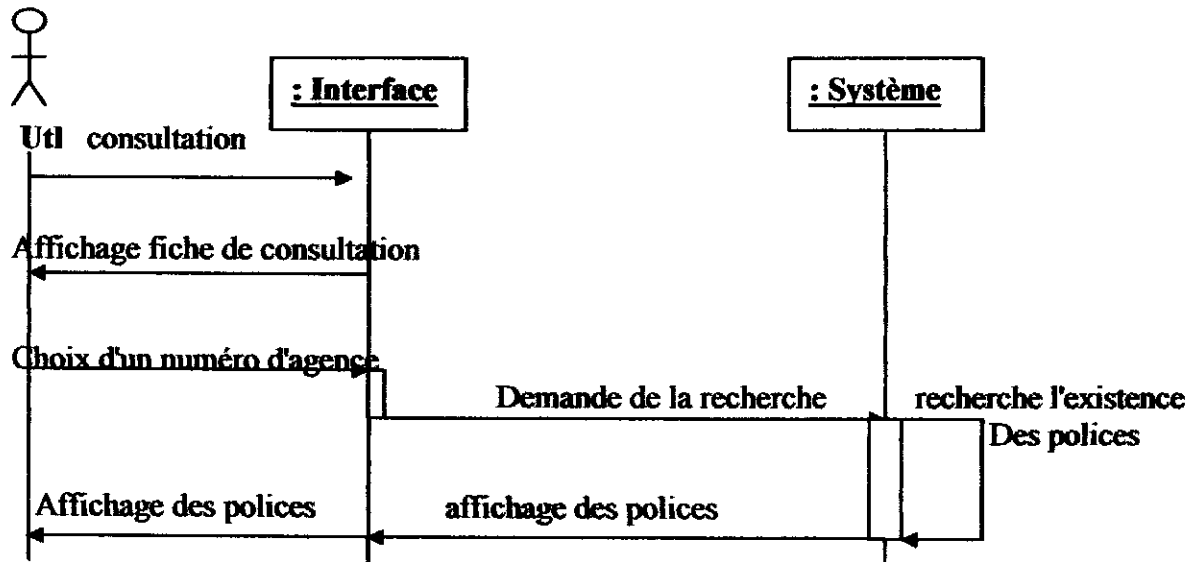


Figure .III.42 diagramme de séquence« consulter polices»

Scénario :

Le producteur d'agence demande la fiche de consultation : consultation, la fiche de consultation apparaît, le producteur sélectionne un numéro d'agence pour consulter les polices de cet agence.

➤ **Consulter avenants/adhérent :**

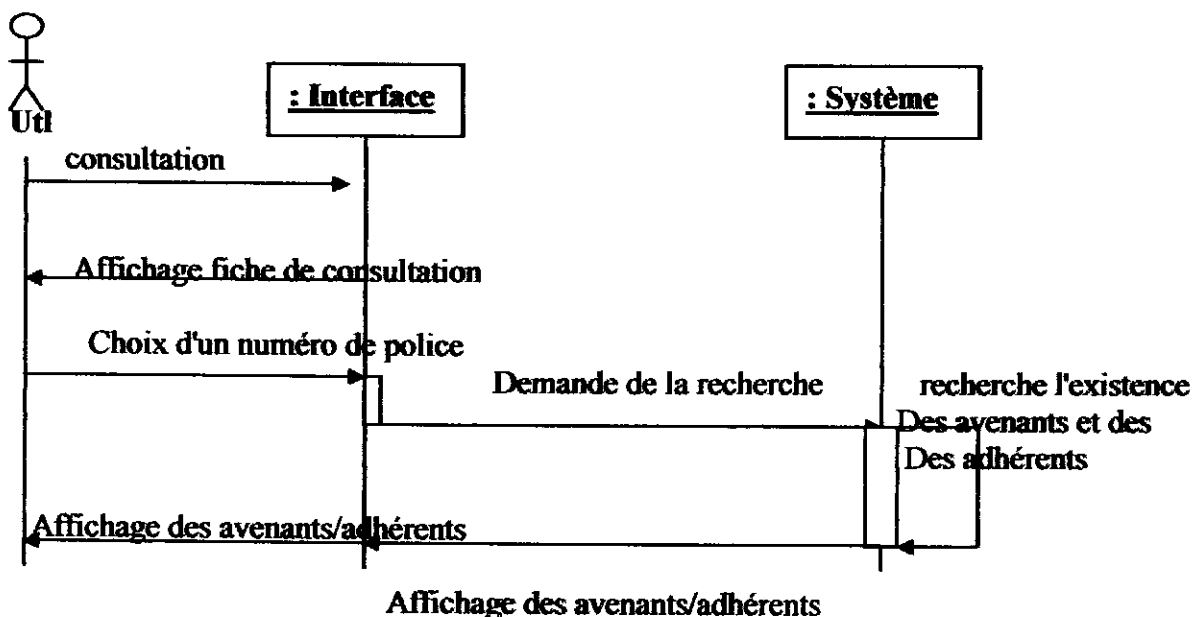


Figure .III.43diagramme de séquence« ajouter avenant»

Scénario :

Le producteur d'agence demande la fiche de consultation : consultation, la fiche de consultation apparaît, le producteur sélectionne un numéro de police pour consulter les avenants et les adhérents de cette police.

➤ **Consulter agences:**

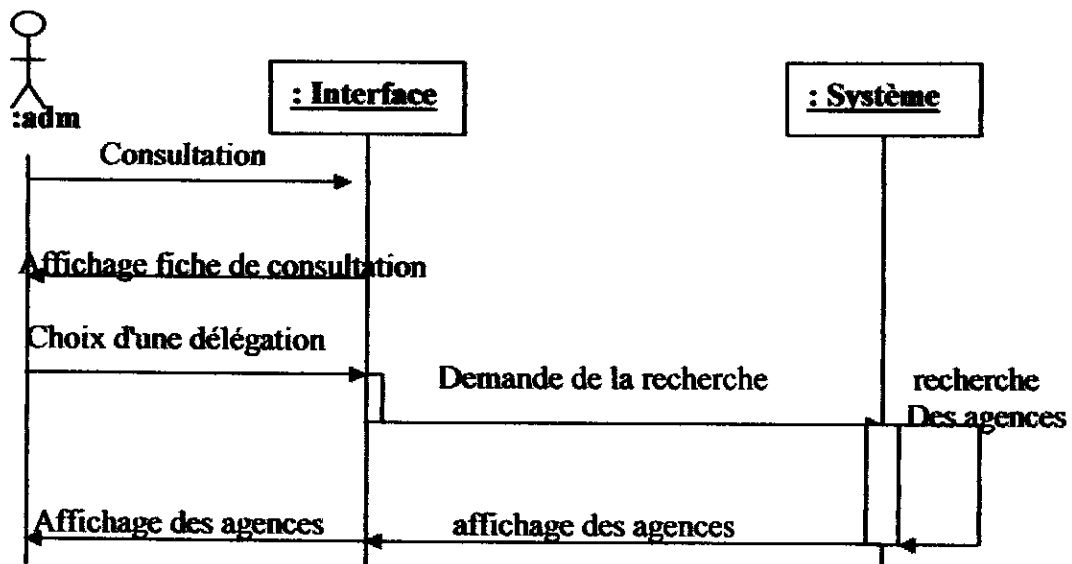


Figure. .III.44 diagramme de séquence« consulter agences»

Scénario :

L'administrateur demande la fiche de consultation : consultation, la fiche de consultation apparaît, L'administrateur sélectionne une délégation pour consulter les agences de cette délégation.

➤ **Consulter remboursement:**

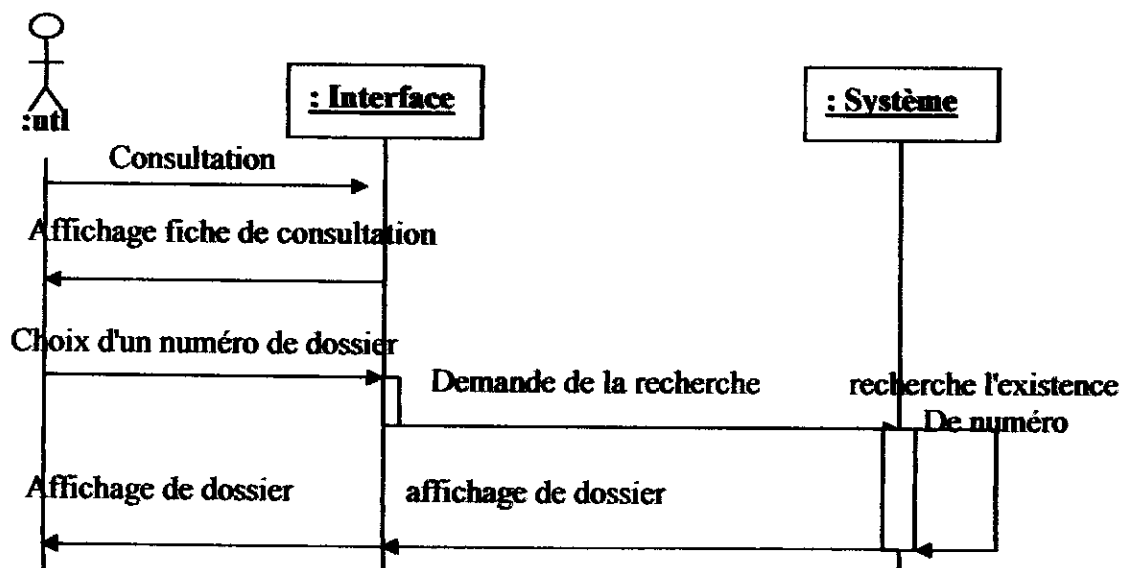


Figure. .III.45.diagramme de séquence« consulter remboursement»

Scénario :

Le producteur d'agence demande la fiche de consultation : consultation, la fiche de consultation apparaît, le producteur sélectionne un numéro de dossier pour le consulter leur.

3. Les diagrammes d'activité :

➤ **Produire police:**

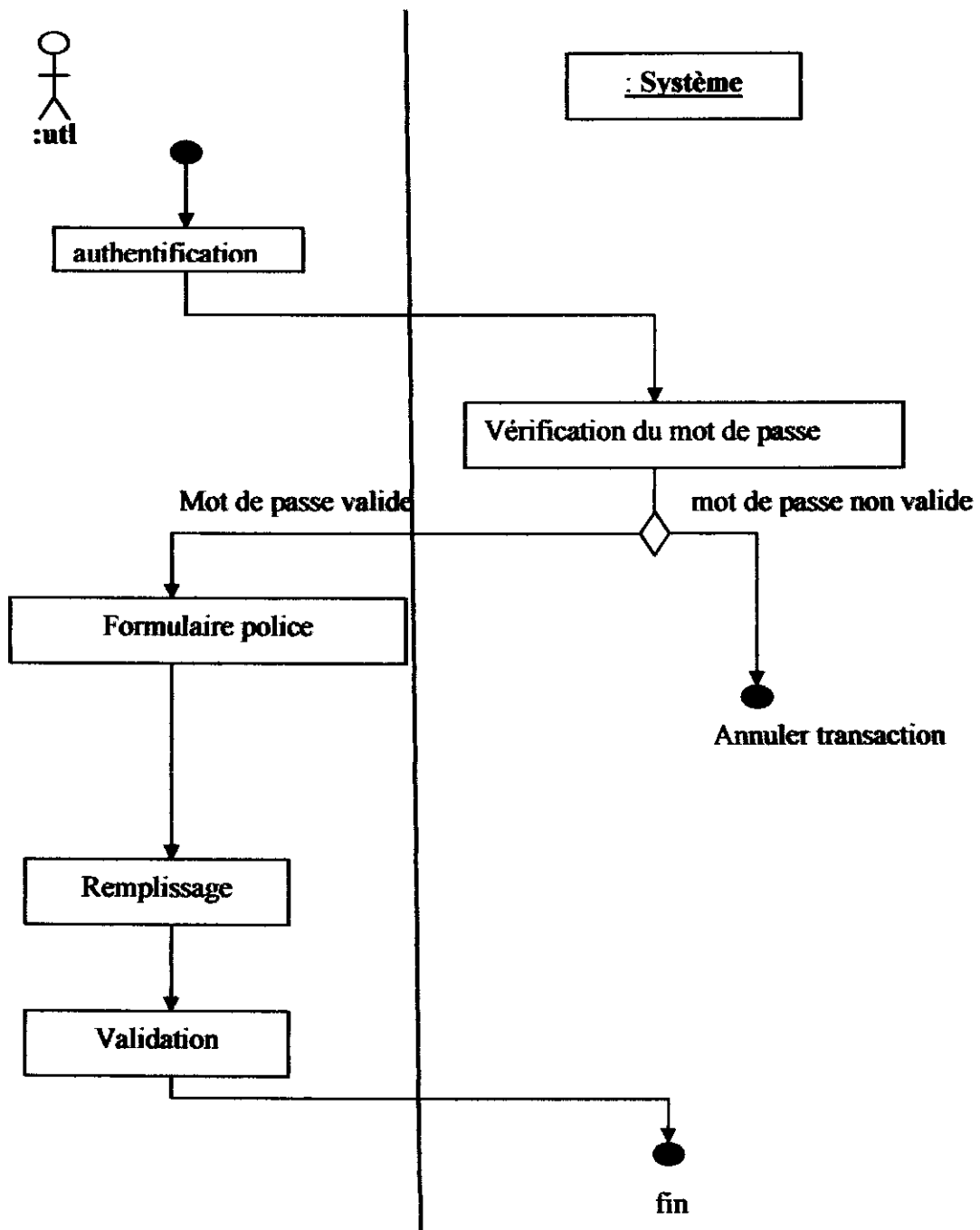


Figure. III.46 diagramme d'activité « produire police »

➤ Produire avenant:

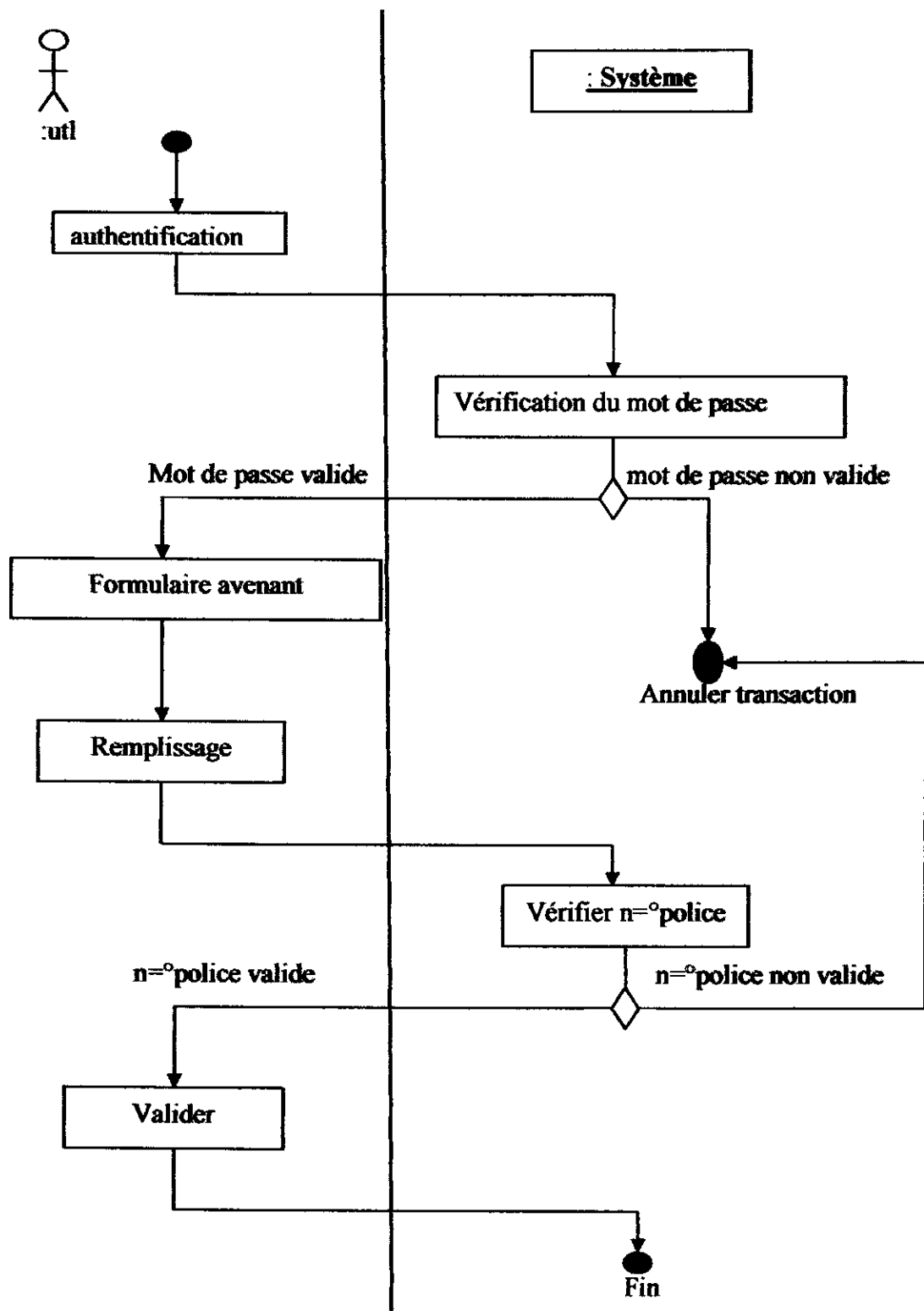


Figure. .III.47 diagramme de activité« produire avenant»

➤ Déclarer sinistre:

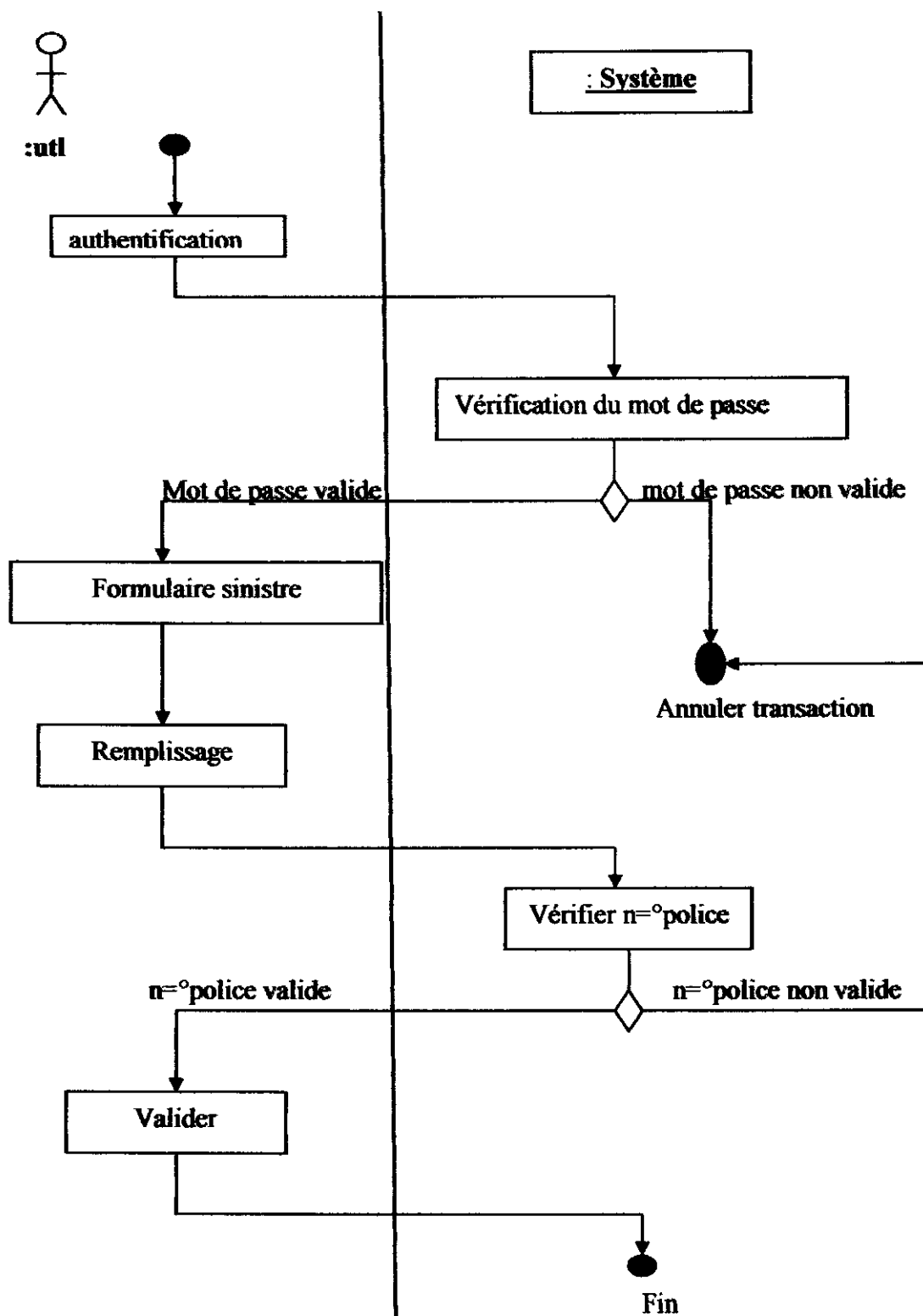


Figure. .III.48 diagramme de activité« déclarer sinistre»

➤ Régler sinistre(remboursement):

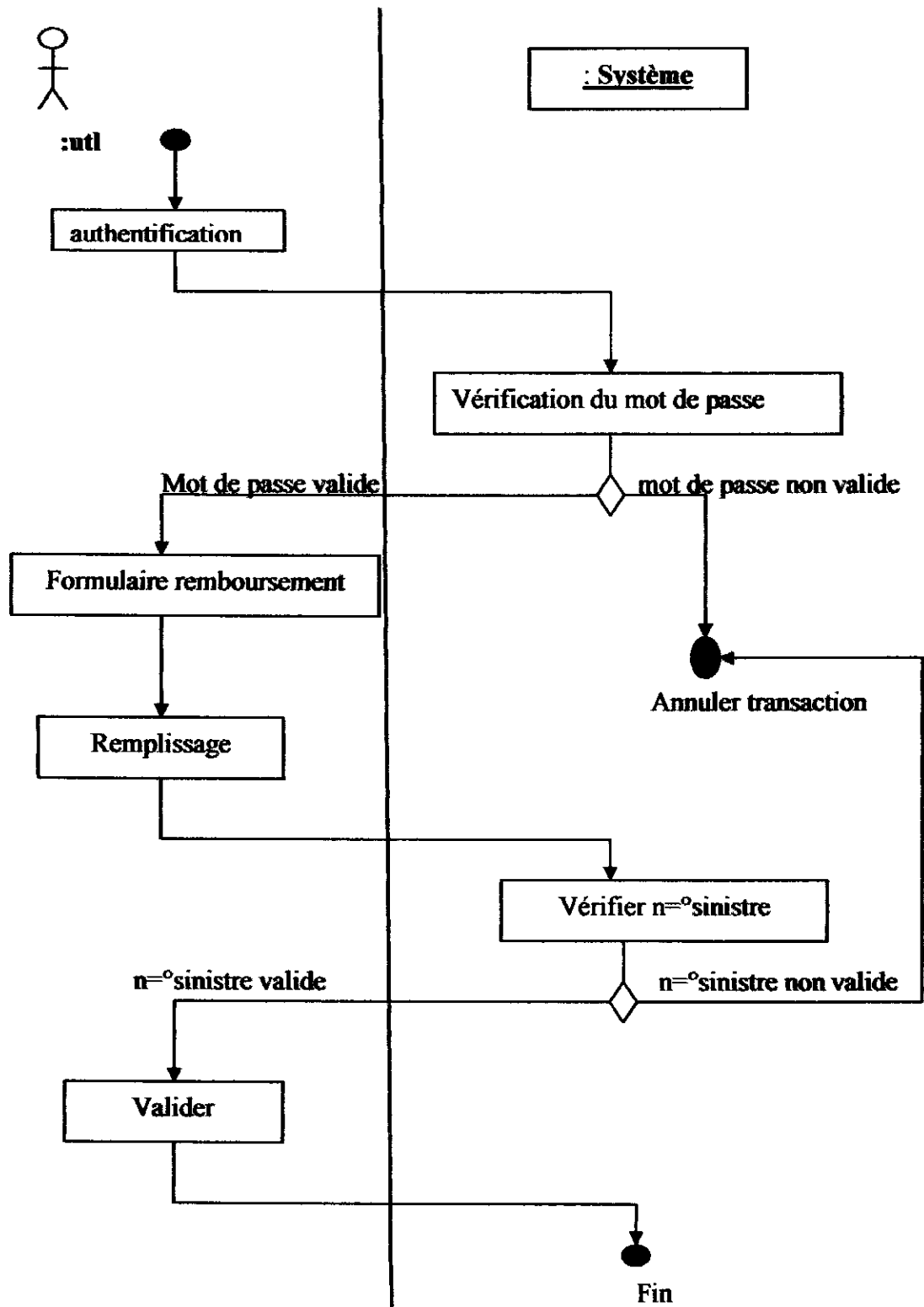


Figure. .III.49 diagramme de activité« remboursement»

➤ Créer délégation:

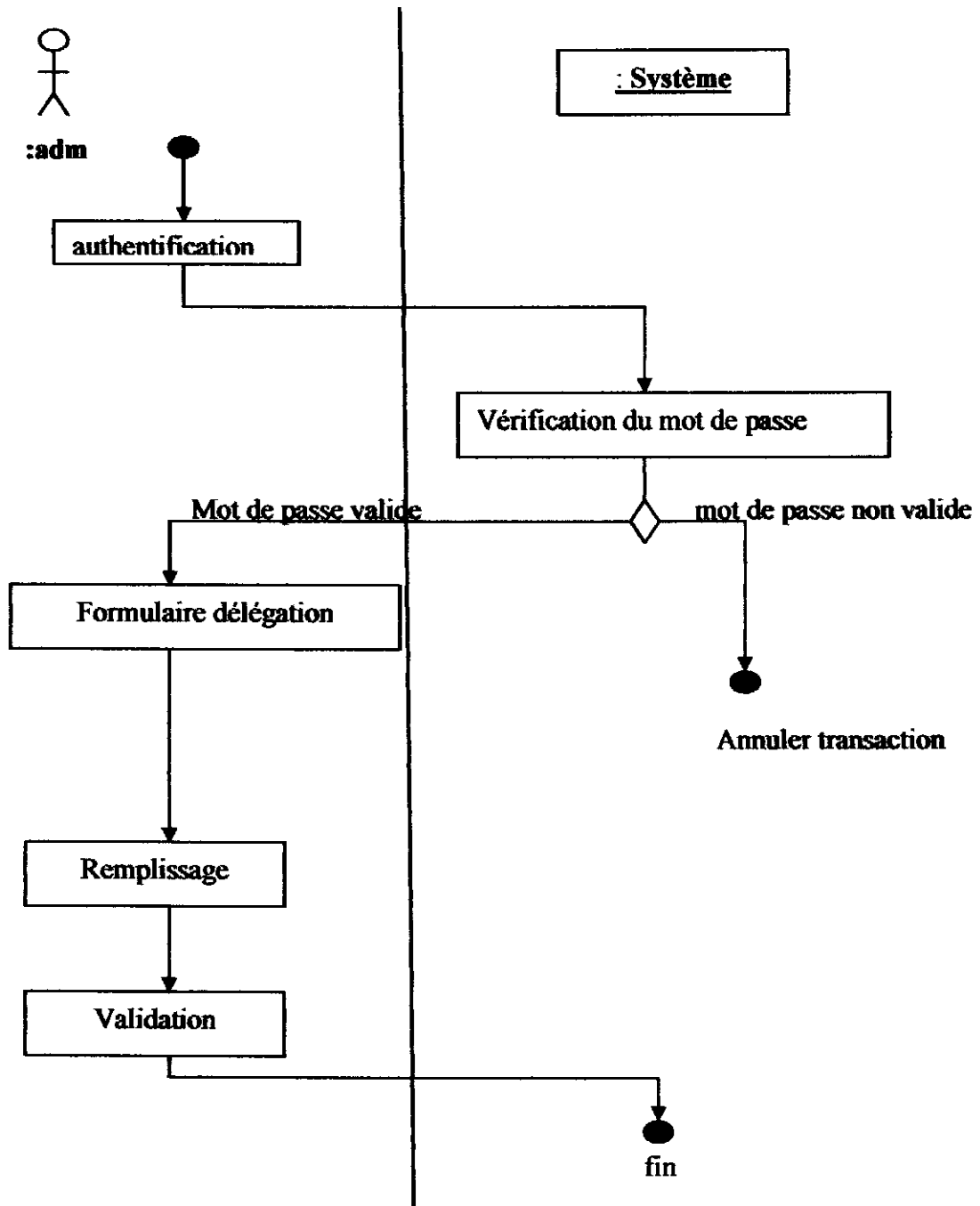


Figure. .III.50 diagramme de activité« créer délégation»

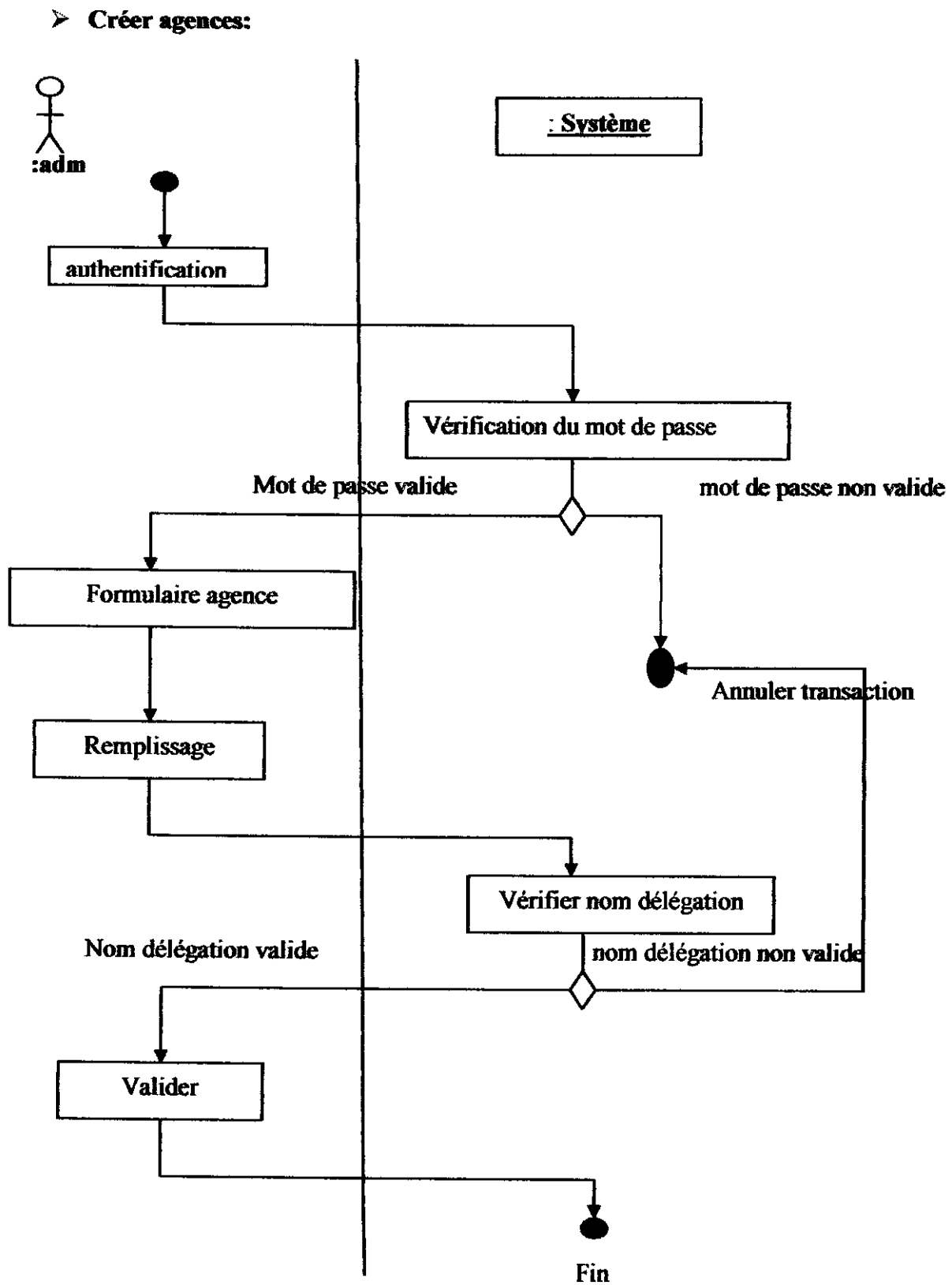


Figure. .III.51 diagramme de activité« créer agence»

➤ Consultation
cas générale: police,avenant,sinistre,remboursement,agence,délégation:

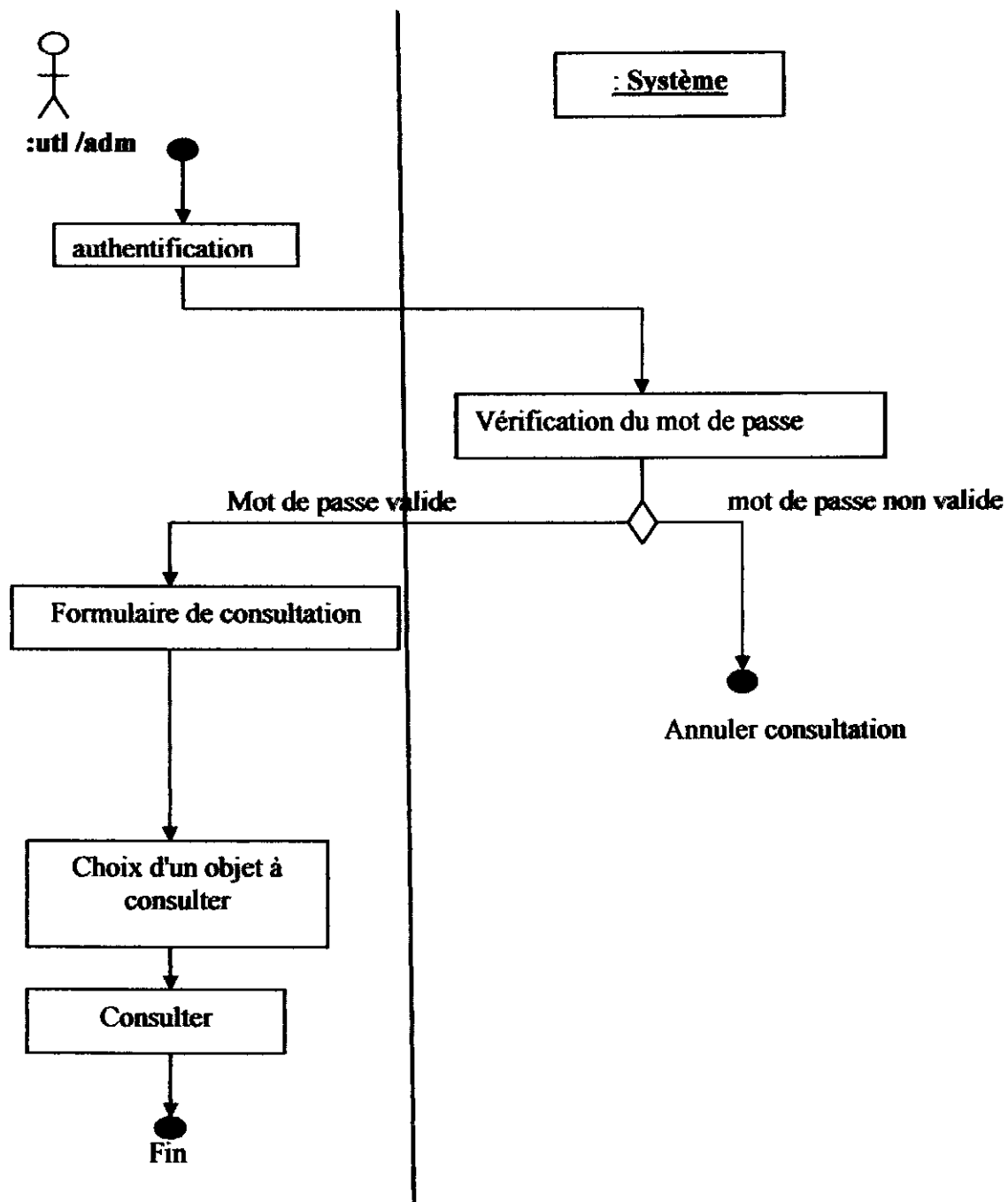


Figure. .III.52 diagramme de activité« consulter»

4. diagrammes de collaboration:

➤ **police :**

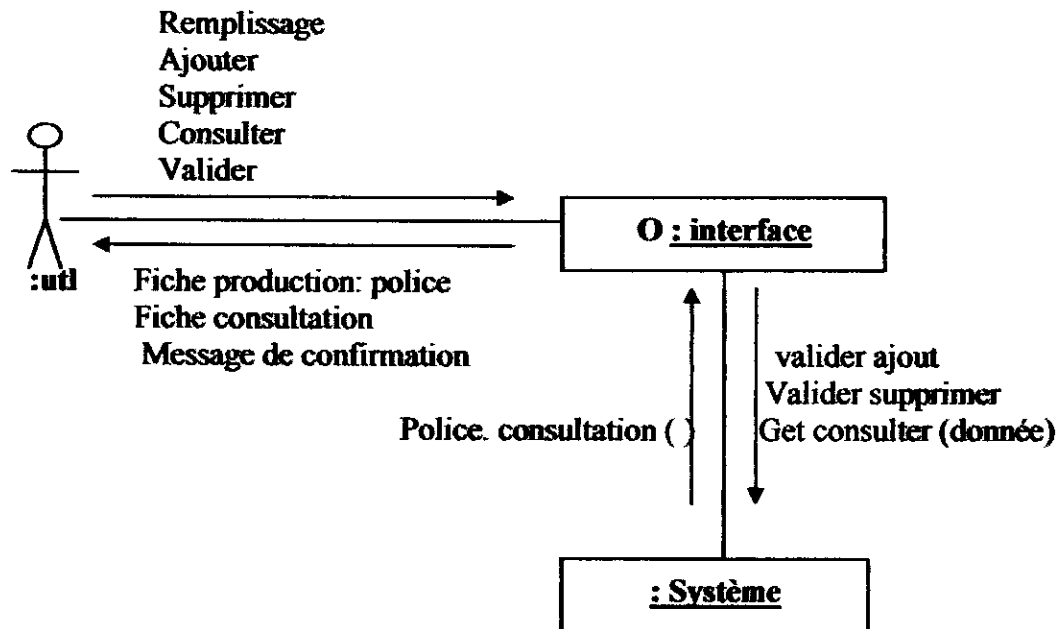


Figure. .III.53 diagramme de collaboration« police»

➤ **avenant :**

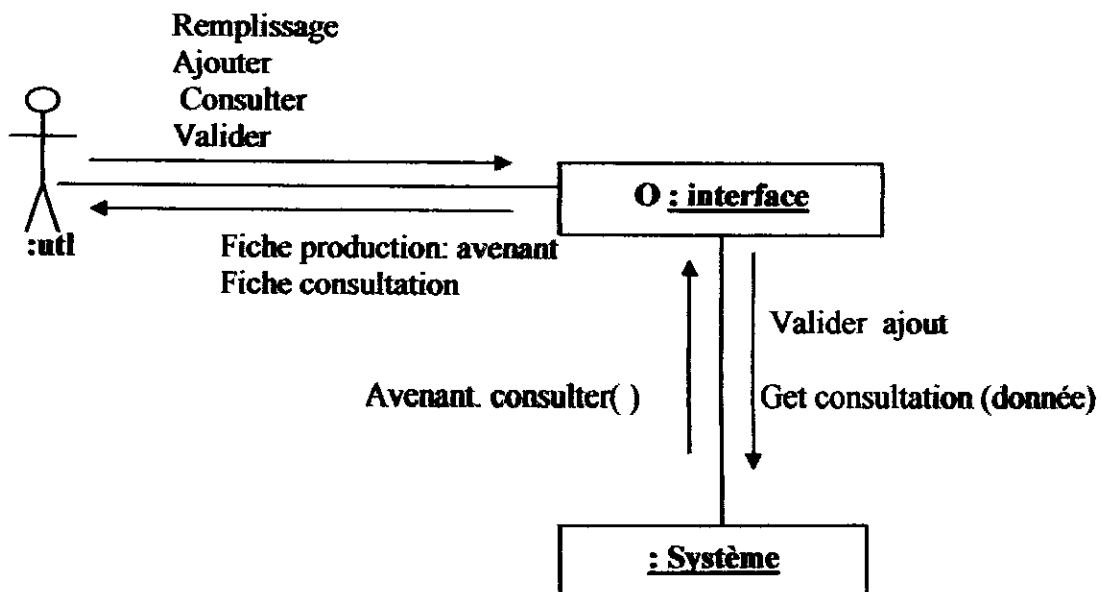


Figure. .III.54 diagramme de collaboration« avenant»

➤ **sinistre :**

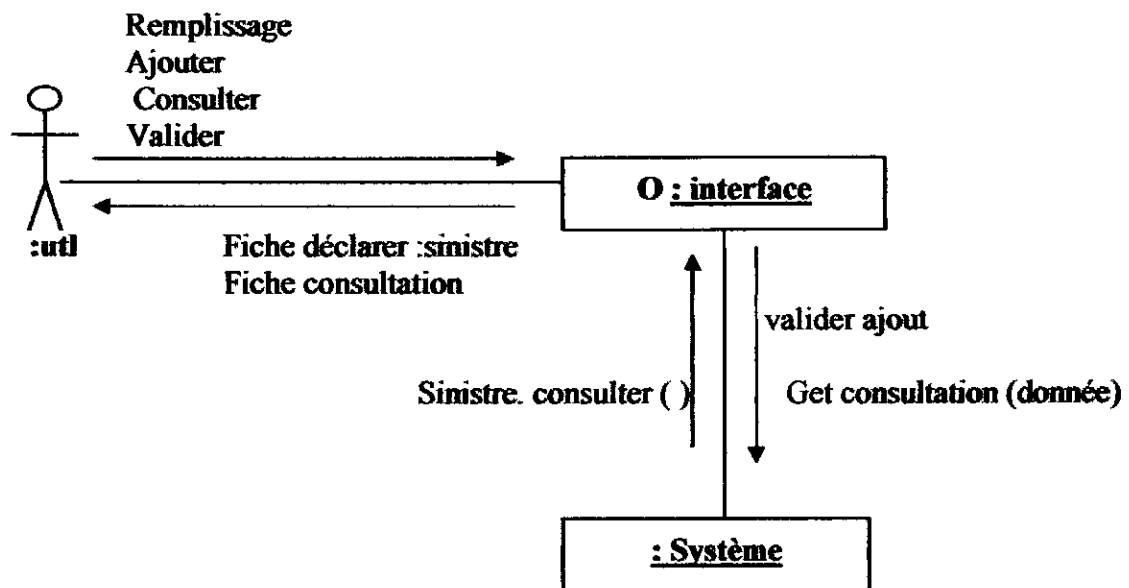


Figure. .III.55 diagramme de collaboration « sinistre »

➤ **remboursement:**

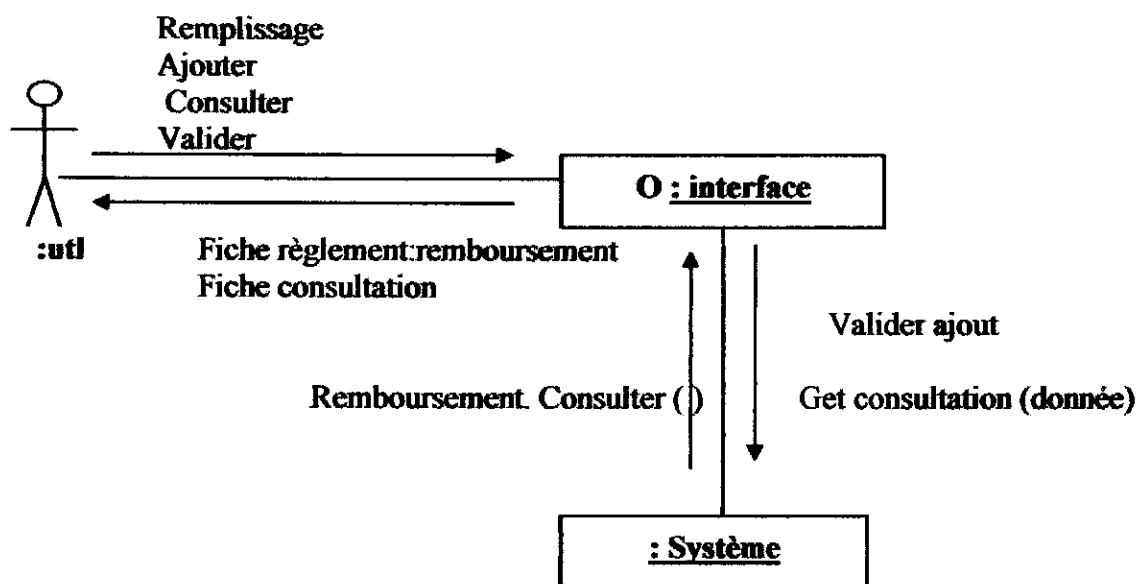


Figure. .III.56 diagramme de collaboration « remboursement »

➤ **adhérent:**

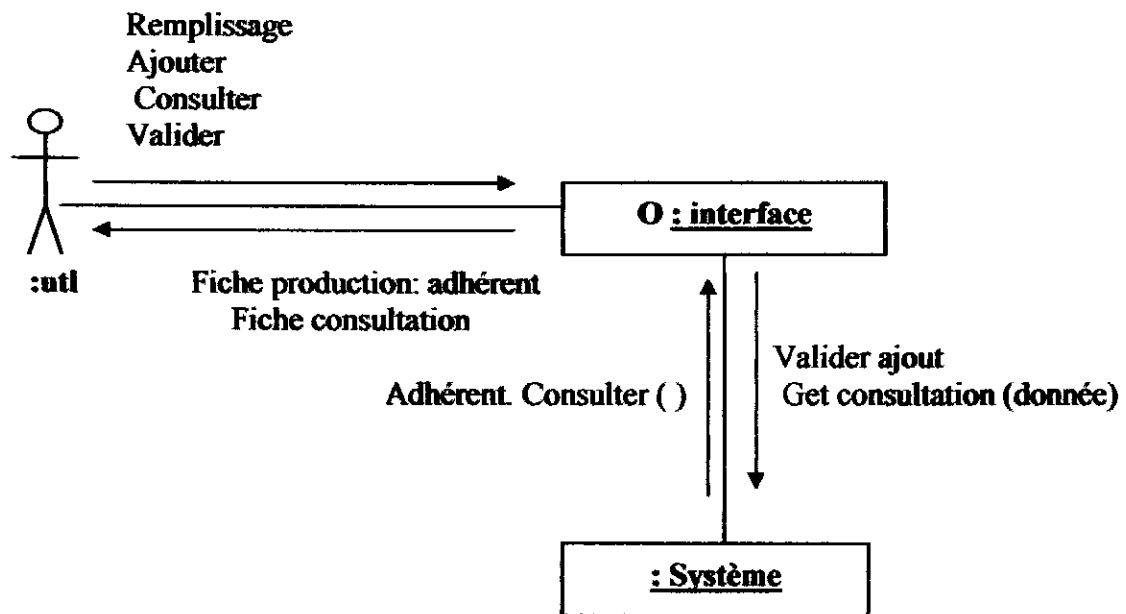


Figure. .III.57 diagramme de collaboration« police»

➤ **Agence**

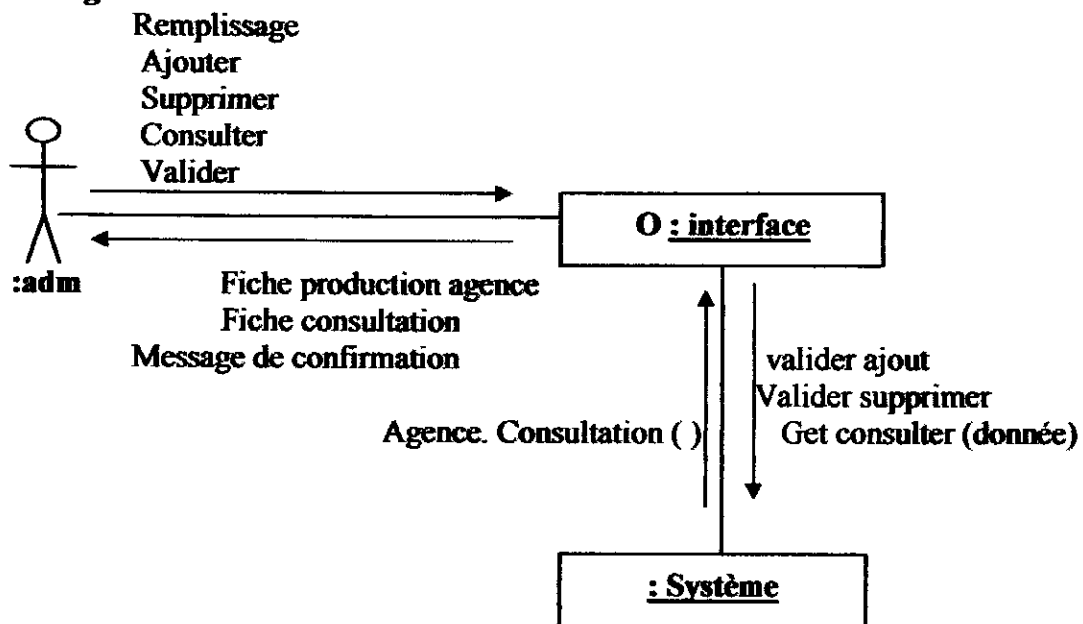


Figure. .III.58 diagramme de collaboration« agence»

➤ **délégation:**

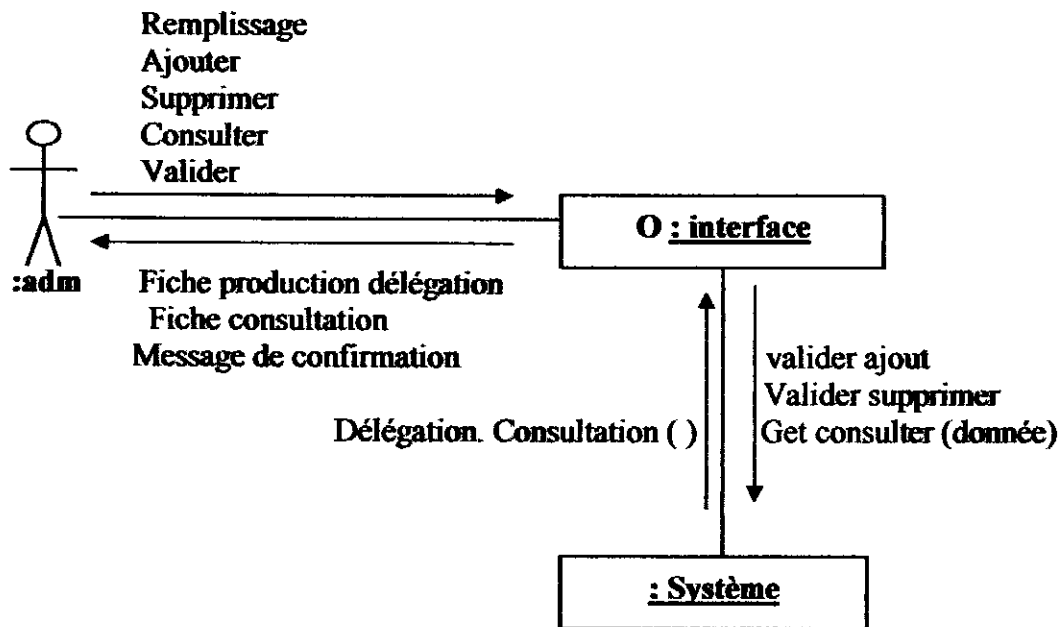


Figure. .III.59diagramme de collaboration« délégation»

III.2. le modèle logique :

1. définition des classes :

classe	définition	identifiant	attributs	Désignation
Délégation	Délégation	cod_del	Cod_del Nom_del Adr	Code délégation Nom délégation Adresse
Agence	Agence	cod_ag	cod_ag Cod_del type Adr	code agence code délégation Type d'agence Adresse
Assuré	La compagnie à assurer	cod_ass	cod_ass Nom Adr Tél Activité Cap_m_ass Cap_s N_eff	code assuré nom adresse téléphone activité professionnelle capital moyen assuré capital social nombre d'effectif
Police	Le contrat	cod_pol	cod_pol cod_ag Cod_ass Date_d Date_f Prime	Numéro police Numéro agence Numéro assurant Date début Date fin Prime
Avenant	Modification subit à un contrat	Cod_av	Cod_av Libélé	Code avenant Libélé avenant
Typ_Avenant	Type avenant	Cod_typav	Cod_pol Cod_av Cod_typav Date_effet Date_echéance Cod_adh Prime	code police code avenant Code type avenant Date effet Date échéance Code adhérent Prime
Garantie	Garantie prise en compte par un contrat	Cod_gar	Code_gar Libellé Type taux	Code garanti libellé type taux de garantie
Typ_Garantie	Type de garantie	cod_pol Cod_gar	cod_pol Cod_gar Libellé Type	code police Code garanti libellé type
Adhérent	Les adhérents	num adh	num adh	Numéro adhérent

	assuré par la police	cod_pol	cod_pol Nom Prénom DdNs N=°ss Montant Ass bénéficiaires Adr Age Nom_conjoint Pre_conjoint Dns_conjoint	Code police nom prénom date de naissance n=° sécurité sociale montant assuré bénéficiaires en cas de décès adresse age nom conjoint prénom conjoint date de naissance du conjoint
sinistre	Les sinistres	cod_sinis	cod_sinis cod_pol Cod_ag cod_ass Date_sinis	code sinistre code police Code agence code assurant Date sinistre
Remboursement	Les sinistres réglés	cod_remb	cod_remb cod_sinis Cod_pol Cod_ag Cod_ass Cod_adh Cod_gar Date_reglement montant	code remboursement Code sinistre Code police Code agence Code sinistre Code adhérent Code garanti Date règlement montant

Tableau III.2 : définition des classes

1.1 présentation de la nouvelle codification:

- **Code agence:** on propose une nouvelle codification ,c'est un numéro séquentiel selon l'ordre de la création.

|0020| → number ;

- **Code police:**

Exemple:

| 0020 | 2006 | / | 001 |

Agence : 4nombres
 année : 4caractère
 / : 1caractère
 N=°séquentiel : 3caractère

Longueur et type : varchar2(12)

- **Code avenant:** on conserve la même codification de l'ancien système

Exemple: |11| → avenant de reconduction

Longueur et type: number(2)

- **Code type_avenant:**

Exemple

|0020| 2006| 0101|

Agence : 4nombres
 année : 4caractère
 / : 1caractère
 N=°séquentiel : 3caractère

Longueur et type : varchar2(12)

- **Code garantie:**

Exemple: |G01|

Longueur et type : varchar2(3)

- **Type garantie:**

Exemple : | B |

B:base
 C: complémentaire

Longueur et type : varchar2(1)

- **Code sinistre :**

Exemple :

0020	2006	/	022
------	------	---	-----

Longueur et type : varchar2(12)

➤ **Code remboursement:**

Exemple:

0020	2006	/	033
------	------	---	-----

Longueur et type : varchar2(8)

➤ **Numéro d'adhérent:**

Exemple:

0001

Longueur et type : number(4)

Agence : 4 nombres

année : 4 caractère

/ : 1 caractère

N=° séquentiel : 3 caractère

Agence : 4 nombres

année : 4 caractère

/ : 1 caractère

N=° séquentiel : 3 caractère

1.2 le diagramme de classe:

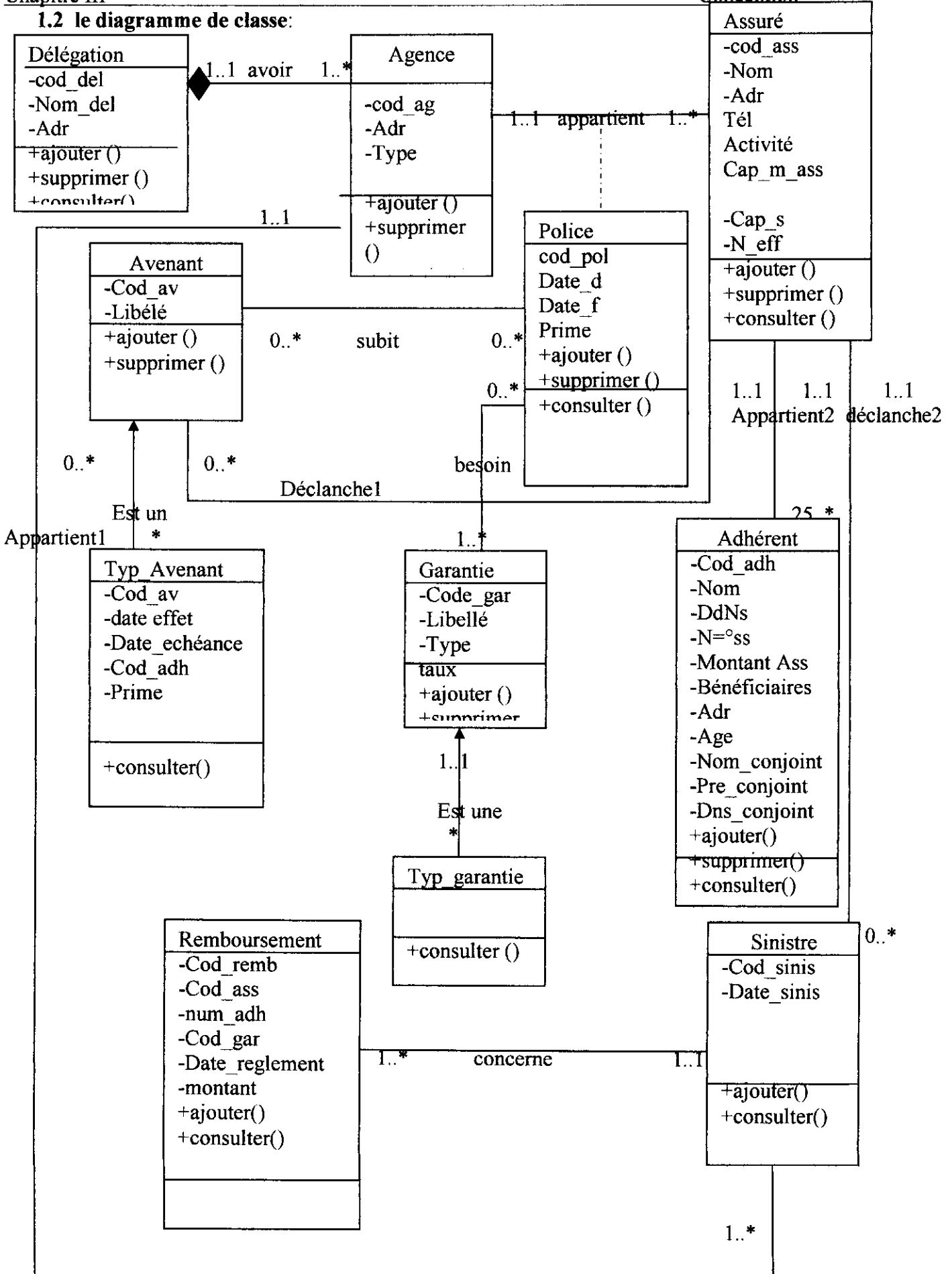
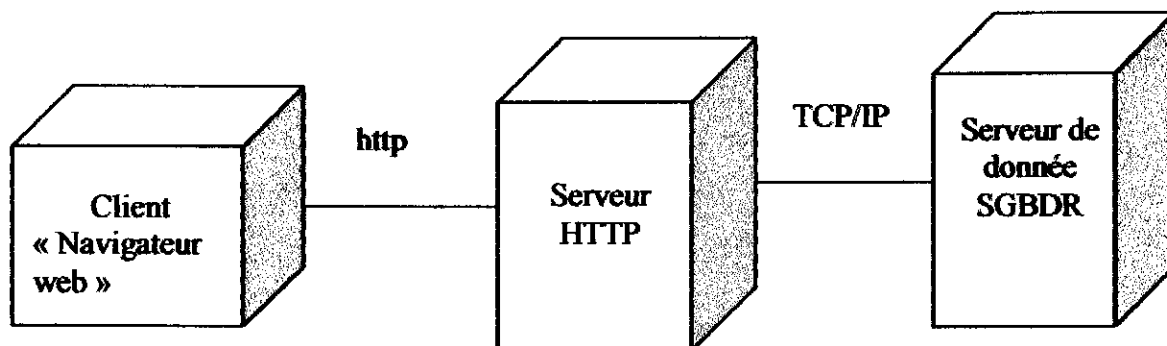


Figure. .III.60 diagramme de classe»

III.3.le modèle physique:**1. le diagramme de déploiement:****Figure. III.61 diagramme de déploiement»**

Chapitre IV

Présentation de l'architecture 3- tiers

I. Introduction :

Dans ce chapitre nous allons exposer la notion des systèmes d'informations distribués et l'architecture trois-tiers ainsi que les bases de données & web.

L'objectif premier d'un système d'information quel qu'il soit est de permettre à plusieurs utilisateurs d'accéder aux mêmes informations. Pour cela il faut donc regrouper les informations utilisées par l'entreprise. En terme technique, cela se traduit par la centralisation des données au sein d'une base de données. L'évolution des systèmes d'information s'est donc basée sur une meilleure subdivision entre les tâches à réaliser pour permettre l'exploitation de ces données par les utilisateurs finaux. Ceci permet de structurer plus efficacement les informations ce qui entraîne à la fois une meilleure organisation de l'entreprise et une meilleure efficacité technique. Cette subdivision a été facilitée par l'avènement des technologies orientées objets qui s'appliquent aussi bien au modèle client-serveur qu'au modèle Internet. Ces technologies permettent une séparation entre les différents composants du système. Il devient alors possible de réaliser de nouvelles architectures permettant la mise à disposition des informations sous différentes formes tout en diminuant les temps de développement. Ces technologies permettent également de faire collaborer une grande diversité de systèmes. On parle alors d'architecture distribuée. Il est ainsi possible de présenter des données en provenance d'un mainframe mélangées à des données en provenance d'un SGBDR, le tout étant affiché dans un browser sur la même page HTML.

II. Les trois niveaux d'abstraction d'une application informatique :

En règle générale, une application informatique peut être découpée en trois niveaux d'abstraction distincts :

1. **la couche de présentation**, encore appelée IHM, permet l'interaction de l'application avec l'utilisateur. Cette couche gère les saisies au clavier, à la souris et la présentation des informations à l'écran. Dans la mesure du possible, elle doit être conviviale et ergonomique.
2. **la logique applicative, les traitements**, décrivant les travaux à réaliser par l'application. Ils peuvent être découpés en deux familles :
 - **les traitements locaux**, regroupant les contrôles effectués au niveau du dialogue avec l'IHM, visant essentiellement le contrôle et l'aide à la saisie,
 - **les traitements globaux**, constituant l'application elle-même. Cette couche, appelée *Business Logic* ou couche métier, contient les règles internes qui régissent une entreprise donnée.
3. **les données**, ou plus exactement l'accès aux données, regroupant l'ensemble des mécanismes permettant la gestion des informations stockées par l'application.

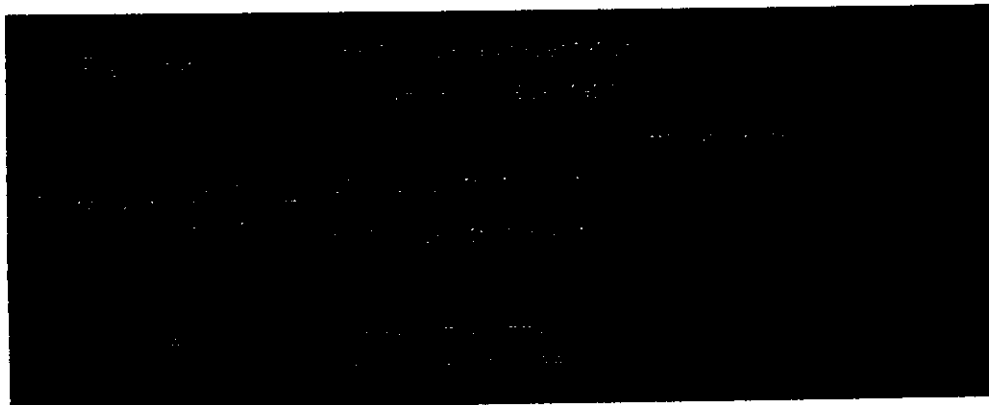


Figure IV.1: Les trois niveaux d'une application informatique

Ces trois niveaux peuvent être imbriqués ou répartis de différentes manières entre plusieurs machines physiques.

Le noyau de l'application est composé de la logique de l'affichage et la logique des traitements. Le découpage et la répartition de ce noyau permettent de distinguer les architectures applicatives suivantes :

- l'architecture 1-tiers,
- l'architecture 2-tiers,
- l'architecture 3-tiers,
- les architectures n-tiers.

Avant la présentation de l'architecture 3-tiers, nous présentons tout d'abord l'architecture un tiers et deux tiers particulièrement les caractéristiques et les limitations de chaque architecture.

III. L'architecture un tiers :

III.1. Présentation

Dans une application un tiers, les trois couches applicatives sont intimement liées et s'exécutent sur le même ordinateur. On ne parle pas ici d'architecture client-serveur, mais d'informatique centralisée.

Dans un contexte multi-utilisateurs, on peut rencontrer deux types d'architecture mettant en oeuvre des applications un tiers :

- des applications sur site central,
- des applications réparties sur des machines indépendantes communiquant par partage de fichiers.

A). Les solutions sur site central (*mainframe*) :

Historiquement, les applications sur site central furent les premières à proposer un accès multi-utilisateurs. Dans ce contexte, les utilisateurs se connectent aux applications exécutées par le serveur central (le *mainframe*) à l'aide de terminaux passifs se comportant en esclaves. C'est le serveur central qui prend en charge l'intégralité des traitements, y compris l'affichage qui est simplement déporté sur des terminaux passifs.

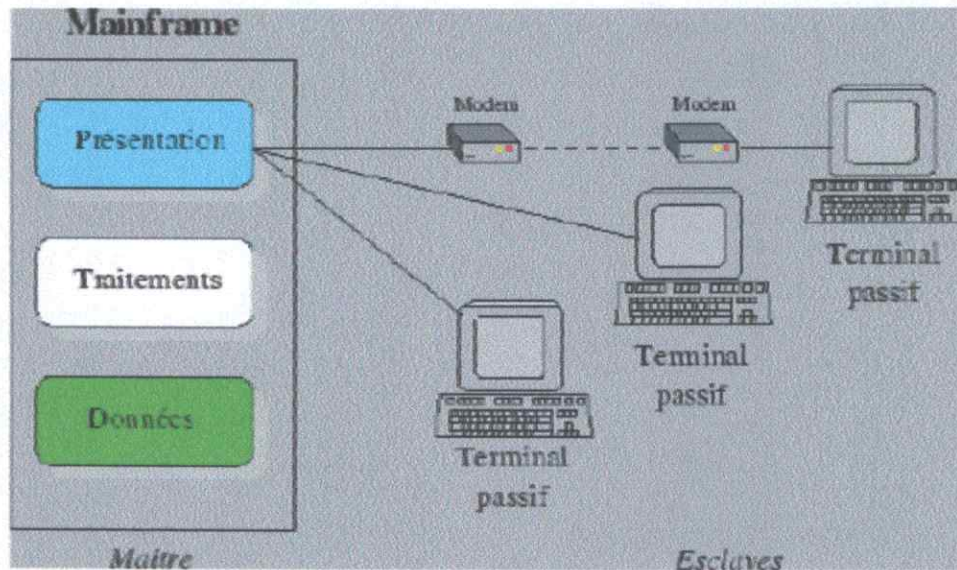


Figure IV.2 : Architecture d'une application sur site central

Ce type d'organisation brille par sa grande facilité d'administration et sa haute disponibilité. Il bénéficie aujourd'hui d'une large palette d'outils de conception, de programmation et d'administration ayant atteint un niveau de maturité et de fiabilité leur permettant encore de soutenir la comparaison avec des solutions beaucoup plus modernes. De plus, la centralisation de la puissance sur une seule et même machine permet une utilisation optimale des ressources.

B). Les applications un tiers déployées :

Avec l'arrivée dans l'entreprise des premiers PC en réseau, il est devenu possible de déployer une application un tiers sur plusieurs ordinateurs indépendants.

Ce type de programme est simple à concevoir et à mettre en oeuvre. Il existe pour cela de nombreux environnements de développement (*dBase, Ms Access, Lotus Approach, Paradox...*) qui sont souvent intégrés aux principales suites bureautiques. L'ergonomie des applications mises en oeuvre, basée sur celle des outils bureautiques, est très riche.

Ce type d'application peut être très satisfaisant pour répondre aux besoins d'un utilisateur isolé et sa mise en oeuvre dans un environnement multi-utilisateur est envisageable. Dans ce contexte, plusieurs utilisateurs se partagent des fichiers de données stockés sur un serveur commun. Le moteur de base de données est exécuté indépendamment sur chaque poste client. La gestion des conflits d'accès aux données doit être prise en charge par chaque programme de façon indépendante, ce qui n'est pas toujours évident. Lors de l'exécution d'une requête, l'intégralité des données nécessaires doit transiter sur le réseau et on arrive vite à saturer ce dernier. De plus, la cohabitation de plusieurs moteurs de base de données indépendants manipulant les mêmes données peut devenir assez instable. Il n'est pas rare de rencontrer des conflits lors de la consultation ou de la modification simultanée d'un même enregistrement par plusieurs utilisateurs. Ces conflits peuvent altérer l'intégrité des données. Enfin, il est difficile d'assurer la confidentialité des données. Ce type de solution est donc à réserver à des applications non critiques exploitées par de petits groupes de travail (une dizaine de personnes au maximum).

III.2.Limitations l'architecture un tiers :

On le voit, les applications sur site central souffrent d'une interface utilisateur en mode caractères et la cohabitation d'applications micro exploitant des données communes n'est pas fiable au delà d'un certain nombre d'utilisateurs.

Il a donc fallu trouver une solution conciliant les avantages des deux premières :

- la fiabilité des solutions sur site central, qui gèrent les données de façon centralisée,
- l'interface utilisateur moderne des applications sur micro-ordinateurs.

Pour obtenir cette synthèse, il a fallu scinder les applications en plusieurs parties distinctes et coopérantes :

- bouleverser gestion centralisée des données,
- gestion locale de l'interface utilisateur.

Ainsi est né le concept du client-serveur.

IV. L'architecture deux tiers :

IV.1. Présentation :

Dans une architecture deux tiers, encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé.

Ce type d'application permet de tirer partie de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui restent gérées de façon centralisée.

La gestion des données est prise en charge par un SGBD centralisé, s'exécutant le plus souvent sur un serveur dédié.

Ce dernier est interrogé en utilisant un langage de requête qui, le plus souvent, est SQL.

Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes. Ce dialogue nécessite l'instauration d'une communication entre client et serveur. Nous allons étudier de quoi elle se compose.

IV.2 Le dialogue client-serveur :

Le modèle client-serveur met en oeuvre une conversation entre deux programmes que l'on peut opposer à l'échange figé "maître-esclave" qu'entretiennent les applications sur site central avec leurs terminaux passifs.

Dans une conversation client-serveur, on distingue donc les deux parties suivantes :

- **Le client**, c'est le programme qui provoque le dialogue,
- **Le serveur**, c'est le programme qui se contente de répondre au client.



Figure IV.3:dialogue entre client et serveur

Le client provoque l'établissement d'une conversation afin de d'obtenir des données ou un résultat de la part du serveur.

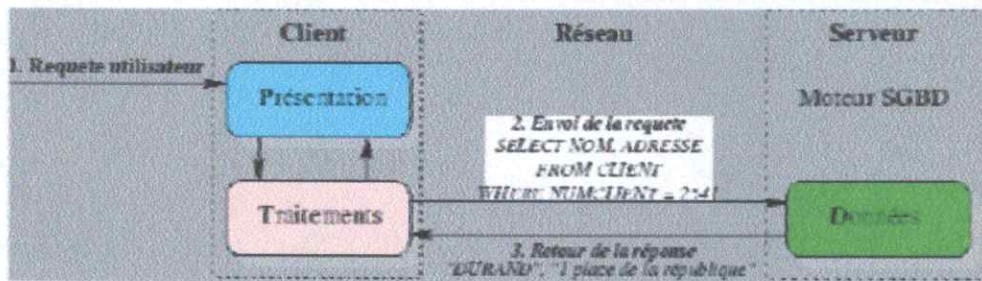


Figure IV.4 : Accès aux données en mode deux tiers

Cet échange de messages transite à travers le réseau reliant les deux machines. Il met en oeuvre des mécanismes relativement complexes qui sont, en général, pris en charge par un *middleware*.

Le Middleware :

▪ **Définition :**

On appelle *middleware*, littéralement "élément du milieu", l'ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différents composants d'une application répartie. Ce dialogue se base sur un protocole applicatif commun, défini par l'API du *middleware*.

Le *middleware* est définie comme une interface de communication universelle entre processus. Il représente véritablement la clef de voûte de toute application client-serveur. L'objectif principal du *middleware* est d'unifier, pour les applications, l'accès et la manipulation de l'ensemble des services disponibles sur le réseau, afin de rendre l'utilisation de ces derniers presque transparente.

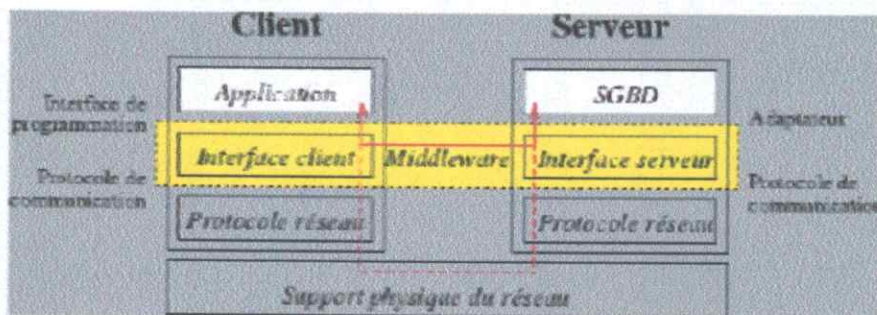


Figure IV.5 : Positionnement du middleware entre client et serveur

▪ **Les services rendus :**

Un *middleware* est susceptible de rendre les services suivants :

- **Conversion** : Service utilisé pour la communication entre machines mettant en oeuvre des formats de données différents, elle est prise en charge par la FAP,
- **Adressage** : Permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès. Dans la mesure du possible, cette fonction doit faire appel aux services d'un annuaire.

- **Sécurité** : Permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations.
- **Communication** : Permet la transmission des messages entre les deux systèmes sans altération. Ce service doit gérer la connexion au serveur, la préparation de l'exécution des requêtes, la récupération des résultats et la dé-connexion de l'utilisateur.

Le middleware masque la complexité des échanges inter-applications et permet ainsi d'élever le niveau des API utilisées par les programmes. Sans ce mécanisme, la programmation d'une application client-serveur serait extrêmement complexe et rigide.

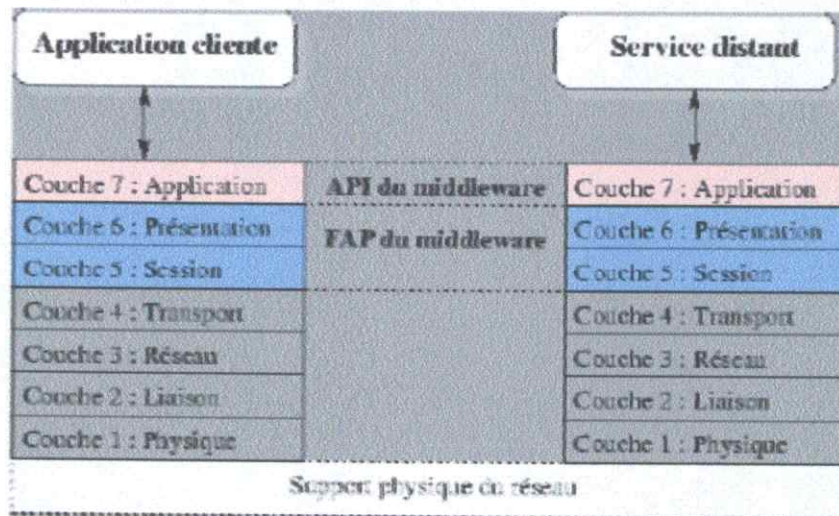


Figure IV.6 : Le middleware par rapport au modèle OSI (Open System Interconnection)

▪ **Exemples de middleware :**

1. **SQL*Net** : Interface propriétaire permettant de faire dialoguer une application cliente avec une base de données Oracle. Ce dialogue peut aussi bien être le passage de requêtes SQL que l'appel de procédures stockées.
2. **ODBC** : Interface standardisée isolant le client du serveur de données. C'est l'implémentation par Microsoft du standard CLI défini par le SQL Access Group. Elle se compose d'un gestionnaire de driver standardisé, d'une API s'interfaçant avec l'application cliente (sous Ms Windows) et d'un driver correspondant au SGBD utilisé.
3. **DCE** : Permet l'appel à des procédures distantes depuis une application.

Le choix d'un middleware est déterminant en matière d'architecture, il joue un grand rôle dans la structuration du système d'information.

Les middleware proposés par les fournisseurs de SGBD sont très performants et permettent de tirer profit de l'ensemble des fonctionnalités du serveur de données pour lequel ils ont été conçus. Par contre, ils ne permettent pas, le plus souvent, l'accès à d'autres sources de données.

Pour certaines applications devant accéder à des services hétérogènes, il est parfois nécessaire de combiner plusieurs middlewares. Dans ce cas, le poste client doit connaître et mettre en oeuvre plusieurs IPC, on en vient à la notion de client lourd.

IV.3.Limites du client-serveur deux tiers : Le client lourd :

L'expérience a démontré qu'il était coûteux et contraignant de vouloir faire porter l'ensemble des traitements applicatifs par le poste client. On en arrive aujourd'hui à ce que l'on appelle le client lourd, ou *fat client*.

L'architecture client-serveur de première génération s'est heurtée à ce constat à l'heure des premiers bilans :

- on ne peut pas soulager la charge du poste client, qui supporte la grande majorité des traitements applicatifs, le poste client est fortement sollicité, il devient de plus en plus complexe et doit être mis à jour régulièrement pour répondre aux besoins des utilisateurs,
- la conversation entre client et serveur est assez bruyante et s'adapte mal à des bandes passantes étroites. De ce fait, ce type d'application est souvent cantonné au réseau local de l'entreprise,
- les applications se prêtent assez mal aux fortes montées en charge car il est difficile de modifier l'architecture initiale,
- la relation étroite qui existe entre le programme client et l'organisation de la partie serveur complique les évolutions de cette dernière,
- ce type d'architecture est grandement rigidifié par les coûts et la complexité de sa maintenance.

Malgré tout, l'architecture deux tiers présente de nombreux avantages qui lui permettent de présenter un bilan globalement positif :

- elle permet l'utilisation d'une interface utilisateur riche,
- elle a permis l'appropriation des applications par l'utilisateur,
- elle a introduit la notion d'interopérabilité.

Pour résoudre les limitations du client-serveur deux tiers tout en conservant ses avantages, on a cherché une architecture plus évoluée, facilitant les forts déploiements à moindre coût. La réponse est apportée par les architectures distribuées.

V. L'architecture trois tiers :

V.1.Système d'information et l'architecture trois-tiers :

Tout système d'information nécessite la réalisation de trois groupes de fonctions: le stockage des données, la logique applicative et la présentation. Ces trois parties sont indépendantes les unes des autres: on peut ainsi vouloir modifier la présentation sans modifier la logique applicative. La conception de chaque partie doit également être indépendante, toutefois la conception de la couche la plus basse est utilisée dans la couche d'au dessus. Ainsi la conception de la logique applicative se base sur le modèle de données, alors que la conception de la présentation dépend de la logique applicative.

V.2.Objectifs :

Les limites de l'architecture deux tiers proviennent en grande partie de la nature du client utilisé :

- le frontal est complexe et non standard (même s'il s'agit presque toujours d'un PC sous Windows),
- le middleware entre client et serveur n'est pas standard.

La solution résiderait donc dans l'utilisation d'un poste client simple communicant avec le serveur par le biais d'un protocole standard.

Dans ce but, l'architecture trois tiers applique les principes suivants :

- les données sont toujours gérées de façon centralisée,
- la présentation est toujours prise en charge par le poste client,
- la logique applicative est prise en charge par un serveur intermédiaire.

V.3.caractéristiques de l'architecture 3-tiers :

L'architecture trois tiers, encore appelée client-serveur de deuxième génération ou client-serveur distribué prend en considération trois principes essentiel :

1. Stockage et accès aux données :

Le système de stockage des données a pour but de conserver une quantité plus ou moins importantes de données de façon structurée. On peut utiliser pour cette partie des systèmes très variés qui peuvent être des systèmes de fichiers, des mainframes, des systèmes de bases de données relationnelles, etc. Le point commun entre tous ces systèmes est qu'ils permettent le partage des données qu'ils contiennent via un réseau. La méthode d'accès à ces données dépendra du type d'organisation de ces données. Dans le cas d'une base de données relationnelle, l'accès peut se faire par des API qui dépendent du langage et de l'environnement. Ainsi en JAVA l'accès se fait via JDBC, alors qu'en C++ il se fera à l'aide d'ODBC. Quel que soit l'API, le langage SQL est utilisé.

2. Logique applicative :

La logique applicative est la réalisation informatique du mode de fonctionnement de l'entreprise. Cette logique constitue les traitements nécessaires sur l'information afin de la rendre exploitable par chaque utilisateur. Les utilisateurs peuvent avoir des besoins très variés et évolutifs. Il devient alors nécessaire de permettre l'évolution du système sans pour autant devoir tout reconstruire. Cette partie utilise les données pour les présenter de façon exploitable par l'utilisateur.

Il convient donc de bien identifier les besoins des utilisateurs afin de réaliser une logique applicative utile tout en structurant les données utilisées.

3. Présentation

La présentation est la partie la plus immédiatement visible pour l'utilisateur. Elle a donc une importance primordiale pour rendre attrayante l'utilisation de l'informatique. Son évolution a été très importante depuis les débuts de l'informatique. Depuis les terminaux en mode texte connectés à des mainframes jusqu'au HTML de nos jours en passant par les applications graphiques développées en client serveur, il y a eu beaucoup de chemin parcouru. Différents types d'interfaces demeurent intéressantes. En effet l'ergonomie d'un site Intranet HTML n'est

pas forcément idéale pour tous les types d'applications. Il peut être intéressant de proposer plusieurs types d'interface pour une seule logique applicative. Par exemple une entreprise disposant d'un site de commerce électronique peut proposer un accès à la liste de ses produits sur Internet en HTML mais disposer d'une interface d'administration réalisée à l'aide d'un applet graphique.

V.4 Principe de l'architecture trois-tiers :

Le principe d'une architecture trois-tiers est relativement simple: il consiste à séparer la réalisation des trois parties vues précédemment (stockage des données, logique applicative, présentation). Nous avons déjà pu entrevoir la possibilité de séparer la conception de ces trois subdivisions, ici il s'agit de séparer leur implantation. Tout comme dans le client-serveur cette séparation signifie qu'il est possible de déployer chaque partie sur un serveur indépendant, toutefois cela n'est pas obligatoire. La mise en place de ce type d'architecture permet dans tous les cas une plus grande évolutivité du système. Il est ainsi possible de commencer par déployer les deux serveurs sur la même machine, puis de déplacer le serveur applicatif sur une autre machine lorsque la charge devient excessive. Les éléments permettant la réalisation classique d'un système en architecture trois tiers sont les suivants:

- système de base de donnée relationnel (SGBDR) pour le stockage des données
- serveur applicatif pour la logique applicative
- navigateur web pour la présentation

Il est important de remarquer que l'essentiel du travail de développement sera implanté au niveau du serveur applicatif. Le SGBDR nécessitera un travail d'administration surtout dans le cas d'une quantité de données importante. Le travail de conception de la base de donnée sera la pierre angulaire du système. En effet l'ensemble du développement s'appuiera sur cette conception. Le navigateur web nécessitera la programmation de code spécifique permettant de gérer l'affichage par ce navigateur. Ce code sera placé sur le serveur applicatif pour permettre une mise à jour sans nécessiter de nouveaux déploiements.

V.5. Les bases de données et le web :

V.5.1 .La révolution Internet :

S'il est un phénomène qui a marqué le monde de l'informatique ces dernières années, c'est bien celui d'Internet.

Ce réseau mondial, créé en 1969 par l'armée américaine, puis utilisé par les chercheurs et autres scientifiques, a connu une croissance phénoménale auprès du grand public avec l'introduction du *World Wide Web* en 1989. Ce dernier permet de publier simplement des informations richement mises en forme et pouvant même, par la suite, contenir des données multimédia.

La véritable révolution du WWW réside dans son caractère universel, rendu possible par l'utilisation de standards reconnus.

V.5.1. 1. Les standards d'Internet :

L'universalité du Web repose sur des standards simples et admis par tous :

- HTML, pour la description des pages disponibles sur le Web,
- HTTP, pour la communication entre navigateur et serveur Web,
- TCP/IP, le protocole réseau largement utilisé par les systèmes Unix,

- CGI, l'interface qui permet de déclencher à distance des traitements sur les serveurs Web.

1.HTML (HyperText Markup Langage) :

HTML est le langage de description de pages hypertexte utilisé par le World Wide Web, il est issu de SGML.

Une page HTML est composée de son contenu propre (du texte plus ou moins richement mis en forme) et de références statiques vers d'autres sources d'informations (images, liens vers d'autres documents...).

La consultation d'une page HTML n'implique donc que très rarement le chargement du seul fichier décrivant la page, il s'accompagne en général de celui de nombreux fichiers annexes. Ces chargements mettent en oeuvre le protocole HTTP.

2. HTTP:

HTTP est un protocole réseau applicatif (dernier niveau du modèle OSI) sans connexion utilisé pour l'échange des données sur le Web. En fait, une connexion HTTP est créée pour chaque requête et ne dure que pendant l'exécution de cette dernière.

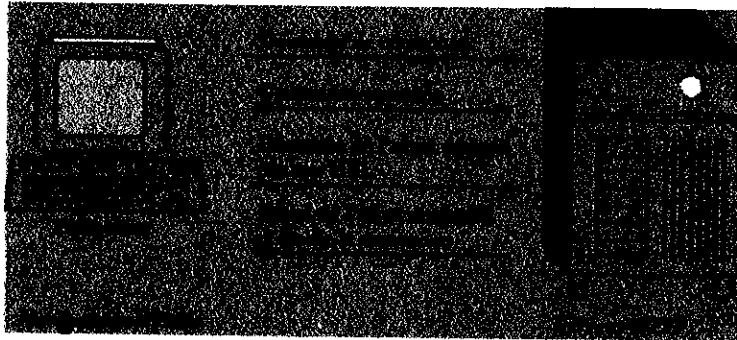


Figure IV.7:Le fonctionnement de base de http

Le protocole HTTP, en temps que protocole réseau applicatif, s'appuie sur un protocole de transport indépendant qui, dans le cadre d'Internet, est TCP/IP.

3.TCP/IP (Transmission Control Protocol / Internet Protocol) :

TCP/IP est un protocole réseau de niveau trois et quatre (réseau et transport) qui s'est largement imposé sur les systèmes Unix, puis sur Internet, et fait aujourd'hui figure de standard universel.

Si le fonctionnement de TCP/IP s'adapte bien à la topologie et à la qualité de service du réseau Internet, on ne peut en dire autant du mariage avec le protocole HTTP. En effet, TCP/IP utilise un mécanisme de "démarrage lent" afin d'éviter les engorgements du réseau. Ce mécanisme permet à la machine émettrice d'ouvrir progressivement une fenêtre de congestion en doublant le nombre de paquets émis à chaque aller-retour. En général, la courte durée des échanges HTTP ne permet pas à la fenêtre de congestion d'atteindre la largeur de bande fournie par le réseau local.

4.CGI (Common Gateway Interface) :

CGI est un standard permettant d'écrire des extensions compatibles avec la grande majorité des serveurs HTTP. Ces extensions permettent l'exécution d'une action par le serveur à la demande d'un client.

Ce mécanisme relativement simple, voire même rustique, entraîne l'exécution d'un processus propre à chaque invocation, ce qui est très consommateur de ressources. De ce fait, des extensions comme ISAPI, NSAPI ou les servlets Java sont souvent préférés au standard CGI.

V.5.1.2. Adaptation à l'entreprise : Intranet

Aucun des mécanismes mis en oeuvre par Internet n'est exempt de défaut et il est relativement simple de trouver plus performant. En fait, la force de l'ensemble repose essentiellement dans son universalité.

La notion d'Intranet est née de l'intégration des principes d'Internet et des technologies déployées dans l'entreprise :

- on utilise le réseau local de l'entreprise,
- les données sont toujours gérées par un SGBD,
- les mécanismes utilisés pour interroger le SGBD sont toujours les mêmes.

V.5.2. Répartition des traitements :

L'architecture trois tiers, encore appelée client-serveur de deuxième génération ou client-serveur distribué, sépare l'application en trois niveaux de service distincts :

- **premier niveau** : l'affichage et les traitements locaux (contrôles de saisie, mise en forme de données...) sont pris en charge par le poste client,
- **deuxième niveau** : les traitements applicatifs globaux sont pris en charge par le service applicatif,
- **troisième niveau** : les services de base de données sont pris en charge par un SGBD.

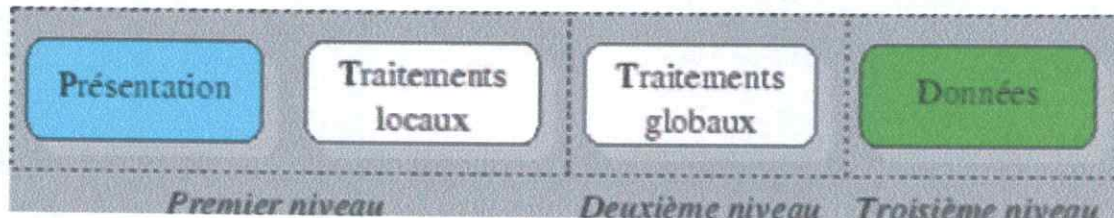


Figure IV.7: Le découpage d'une application en pavés fonctionnels indépendants

Tous ces niveaux étant indépendants, ils peuvent être implantés sur des machines différentes, de ce fait :

- le poste client ne supporte plus l'ensemble des traitements, il est moins sollicité et peut être moins évolué, donc moins coûteux,
- les ressources présentes sur le réseau sont mieux exploitées, puisque les traitements applicatifs peuvent être partagés ou regroupés (le serveur d'application peut s'exécuter sur la même machine que le SGBD),
- la fiabilité et les performances de certains traitements se trouvent améliorées par leur centralisation,
- il est relativement simple de faire face à une forte montée en charge, en renforçant le service applicatif.

Dans le cadre d'un Intranet, le poste client prend la forme d'un simple navigateur Web, le service applicatif est assuré par un serveur HTTP et la communication avec le SGBD met en oeuvre les mécanismes bien connus des applications client-serveur de la première génération.

Ce type d'architecture fait une distinction nette entre deux tronçons de communication indépendants et délimités par le serveur HTTP :

- Le premier tronçon relie le poste client au serveur Web pour permettre l'interaction avec l'utilisateur et la visualisation des résultats. On l'appelle **circuit froid** et n'est composé que de standards (principalement HTML et HTTP). Le serveur Web tient le rôle de "façade HTTP",
- le deuxième tronçon permet la collecte des données, il est aussi appelé **circuit chaud**. Les mécanismes utilisés sont comparables à ceux mis en oeuvre pour une application deux tiers. Ils ne franchissent jamais la façade HTTP et, de ce fait, peuvent évoluer sans impacter la configuration des postes clients.

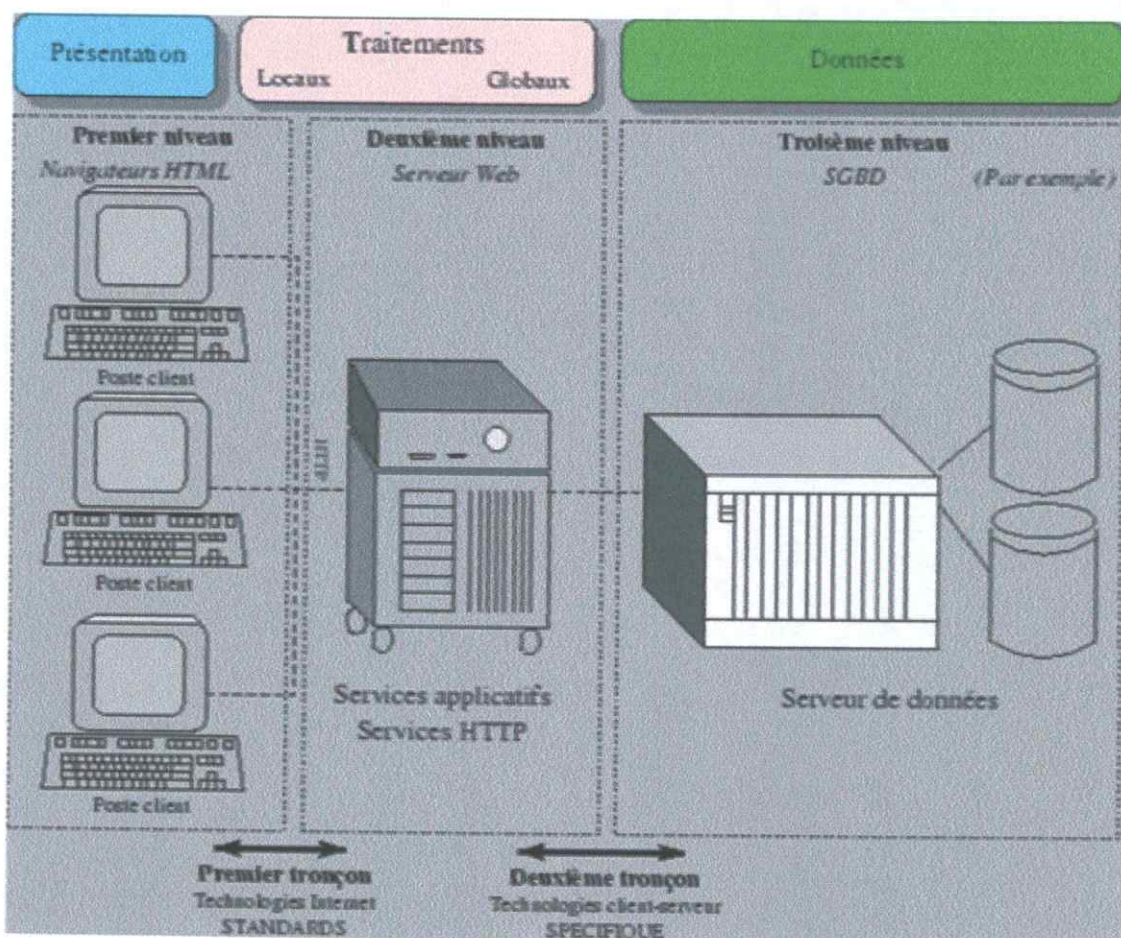


Figure IV.8: Répartition des couches applicatives dans une architecture trois tiers

V.5.3. Le client léger :

V.5.3.1. Présentation

Dans l'architecture trois tiers, le poste client est communément appelé client léger ou *Thin Client*, par opposition au client lourd des architectures deux tiers. Il ne prend en charge que la présentation de l'application avec, éventuellement, une partie de logique applicative permettant une vérification immédiate de la saisie et la mise en forme des données. Il est souvent constitué d'un simple navigateur Internet.

Le poste client ne communique qu'avec la façade HTTP de l'application et ne dispose d'aucune connaissance des traitements applicatifs ou de la structure des données exploitées. Les évolutions de l'application sont donc possibles sans nécessiter de modification de la partie cliente.

Par exemple, un internaute se connectant à *www.yahoo.fr* pour effectuer une recherche provoque, sans même le savoir, l'exécution de traitements sur le serveur. Si ces traitements évoluent, ce qui doit arriver relativement souvent, le client continuera à utiliser le service sans se rendre compte des changements (sauf s'ils lui apportent de nouveaux services).

De plus, ce même internaute peut se connecter au serveur en utilisant tout type de poste client disposant d'un navigateur compatible HTML (PC sous Windows, Macintosh, Station Unix, WebPhone...).

On voit donc ici la force des architectures trois tiers par rapport au client-serveur de première génération. Le déploiement est immédiat, les évolutions peuvent être transparentes pour l'utilisateur et les caractéristiques du poste client sont libres.

V.5.3.2. Ergonomie :

✓ Utilisation d'HTML

Les pages HTML, même avec l'aide de langages de script, sont loin d'atteindre les possibilités offertes par l'environnement Windows :

- le multifenêtrage n'est pas facile à mettre en oeuvre,
- le déroulement de l'application doit se faire séquentiellement,
- les pages affichées sont relativement statiques,
- le développement multi plateforme peut être contraignant.
- l'ergonomie de l'application est limitée aux possibilités du navigateur.

Pour ces raisons, certaines applications ne sont pas réalisables dans une architecture de type Intranet (info centres, applications bureautiques...).

Dans les autres cas, l'appauvrissement de l'interface utilisateur est souvent le gage d'une plus grande facilité de prise en main de l'application. Il est rare en effet de devoir suivre une formation pour apprendre à se servir d'un site Web particulier. La plupart du temps, une formation générale à l'ergonomie des sites Web suffit.

Cette perte de richesse peut donc se transformer en avantage, à condition de respecter une charte graphique et ergonomique cohérente pour toutes les applications Intranet d'une entreprise.

Il est possible d'aller au delà des possibilités offertes par le langage HTML en y introduisant des applets Java ou des contrôles ActiveX. Nous ne parlerons pas ici des modules d'extension du navigateur, encore appelés *plug-in*, qui étaient en vogue avant l'arrivée des solutions Java et ActiveX, car cette solution est trop contraignante à déployer.

✓ Utilisation de Java

Java est un langage de développement orienté objet et multi plateforme introduit par Sun en 1995. Il s'agit de l'adaptation à Internet du langage OAK^{3,8}, initialement étudié pour les environnements de petite taille. Il permet, entre autre, d'écrire de petites applications, appelées *applet*, pouvant être intégrées à des pages HTML pour en enrichir le contenu.

Le caractère multi plateforme de Java se prête bien à une utilisation sur Internet, où les caractéristiques des postes clients ne sont pas maîtrisées. Il repose sur l'utilisation d'un interpréteur de pseudo-code Java, pompeusement appelé "machine virtuelle".

Les programmes Java ne sont pas compilés en code machine, mais en pseudo-code Java uniquement compréhensible par la machine virtuelle. Cette dernière interprète le code et se charge de lier les modules au moment de l'exécution, en les téléchargeant si nécessaire. La liaison dynamique des modules au moment de l'exécution permet d'optimiser le trafic réseau, puisqu'on ne charge que le strict nécessaire.

Pour ces raisons, un programme Java s'exécute plus lentement que son équivalent compilé. La compilation à la volée des programmes Java et plus encore, la technologie d'optimisation dynamique de code *HotSpot* de Sun, permettent de réduire l'écart de performance avec des programmes compilés. Java propose aujourd'hui une large palette de composants graphiques et multimédia qui permettent d'atteindre la richesse fonctionnelle des applications Windows. Une applet Java est aussi capable d'exploiter directement un serveur de données en utilisant JDBC ou de faire appel à des procédures distantes en utilisant RMI ou CORBA.

✓ servlet :

Un conteneur de servlet est un environnement applicatif permettant l'exécution de servlet. Une Servlet est un petit programme Java qui s'exécute côté serveur (à l'opposé d'une applet Java qui s'exécute côté client). Une Servlet a pour fonction d'étendre les fonctionnalités d'un serveur Web. Ils sont par exemple utilisés pour la connexion à une base de données, les transactions bancaires, les sites de commerce électronique...

le conteneur de Servlet va gérer le cycle de vie du Servlet.

Une servlet permet donc de générer un contenu dynamique en réponse à une requête cliente.

Le processus d'exécution d'une Servlet est :

- une requête HTTP cliente appelle une Servlet
- le serveur Web route la requête HTTP vers le conteneur de Servlet
- le conteneur traduit la requête HTTP en une méthode d'invocation Java
- le conteneur transmet la requête au Servlet



- la Servlet génère le contenu dynamique (avec par exemple un appel à une base de données)
 - la Servlet envoie le contenu au client
- ✓ **Page JSP :**

Un JSP est un fichier composé de code HTML et Java. La différence entre une page JSP et une servlet est que la page JSP est composée principalement de code HTML alors que c'est le code Java qui prédomine dans une servlet (une servlet représente la couche métier de l'application). Les JSP sont donc principalement utilisés pour traiter les requêtes clientes et leur fournir une réponse au format HTML.

V.5.3.3. Avantages de l'architecture 3-tiers :

Cette architecture se développe actuellement au sein des entreprises grâce aux nombreux avantages qu'elle présente. Malgré la différence évidente entre une architecture trois tiers et un système client-serveur (l'apparition d'un serveur pour la logique applicative), le système reste basé sur les technologies éprouvées détaillées précédemment (aspect relationnel et transaction). La logique applicative est déplacée au niveau du serveur d'application mais reste programmée à l'aide des mêmes technologies liées aux bases de données relationnelles. En particulier l'utilisation du langage SQL reste jusqu'à présent la solution la plus intéressante au niveau de la qualité logicielle. Elle présente à la fois une grande fiabilité, une bonne disponibilité, une excellente évolutivité, ... Toutefois il faut prendre en compte deux facteurs importants: d'une part le choix du SGBDR (ils n'ont pas tous les même qualités), d'autre part la qualité des programmes utilisant la base de données (aussi bien au niveau de la conception que de la programmation). L'avantage principal d'une architecture multi-tiers est la facilité de déploiement. L'application en elle même n'est déployée que sur la partie serveur (serveur applicatif et serveur de base de données). Le client ne nécessite qu'une installation et une configuration minime. En effet il suffit d'installer un navigateur web compatible avec l'application pour que le client puisse accéder à l'application, ce navigateur étant par ailleurs souvent installé par défaut sur toutes les machines. Cette facilité de déploiement aura pour conséquence non seulement de réduire le coût de déploiement mais aussi de permettre une évolution régulière du système. Cette évolution ne nécessitera que la mise à jour de l'application sur le serveur applicatif. Ceci est très important car cette évolutivité est un des problèmes majeurs de l'informatique. Le troisième avantage est l'amélioration de la sécurité. Dans un système client-serveur tous les clients accédaient à la base de données ce qui la rendait vulnérable. Avec une architecture multi-tiers l'accès à la base n'est effectué que par le serveur applicatif. Ce serveur est le seul à connaître la façon de se connecter à cette base. Il ne partage aucune des informations permettant l'accès aux données, en particulier le login et le password de la base. Il est alors possible de gérer la sécurité au niveau de ce serveur applicatif, par exemple en maintenant la liste des utilisateurs avec leurs mots de passe ainsi que leurs droits d'accès aux fonctions du système. On peut même améliorer encore la sécurité par la mise en place d'une architecture réseau interdisant totalement l'accès au serveur de base de données pour les utilisateurs finaux. La mise en place de firewall correctement configuré permettra ceci. L'avantage le plus intéressant est sans aucun doute l'évolutivité du système. En effet les besoins actuels sont relativement vastes, toutefois la réalisation totale semble relativement longue et difficile à réaliser d'un seul coup. De plus des problèmes risquent de nécessiter une mise à jour répétée du système. Un autre aspect intéressant est la possibilité d'utiliser le système en Extranet,

V.5.3.4. Conclusion :

Etant donné que notre travail est basé sur les bases de données et le Web nous avons mis l'accent sur l'architecture trois-tiers et ses trois niveaux ainsi que l'environnement logiciel qui permettent a cette architecture de fonctionné sur le Web, pour la conception d'applications web et des pages dynamiques.

Chapitre viimplémentation

I. les outils de développement :

Pour atteindre les objectifs fixés en début de stage en matière d'environnement cible pour l'assurance groupe, nous nous sommes intéressés à différents environnements de développement. Notre choix s'est porté sur la plate-forme de développement 100%oracle.

Ce choix a été déterminé par le fait que l'application doit utiliser un SGBD pour le traitement de données et être disponible et intégré dans les navigateurs web existant sur le marché. L'architecture physique de notre nouveau système est présentée par la figure ci dessous :

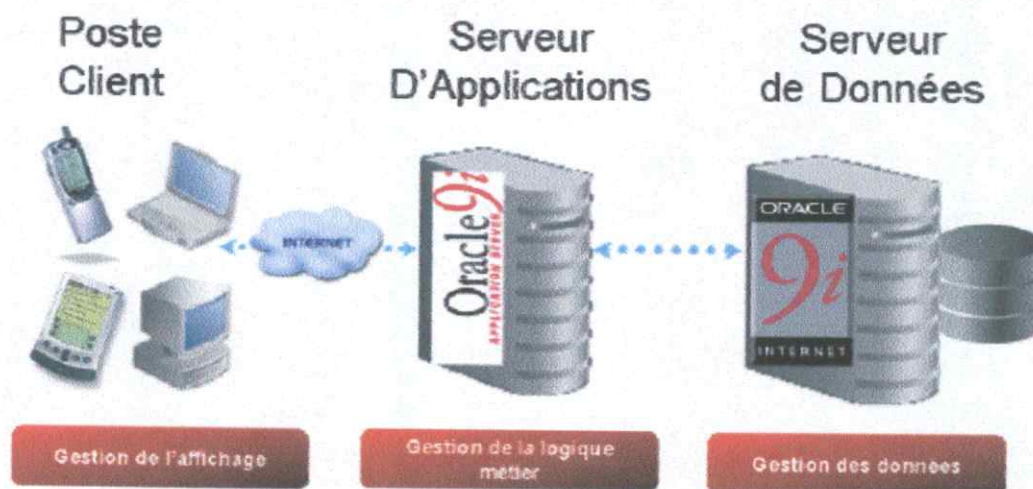


Figure V.1 Architecture physique du système

La plate-forme Oracle s'appuie sur une architecture à trois niveaux :

- Le Poste Client qui assure la gestion de l'affichage avec un simple navigateur Web,
- Le Serveur d'Applications qui gère la logique métier est basé sur Oracle9iAS,
- Le Serveur de Données où est stockée la base de données sur Oracle9i et qui gère les transactions orientées données.

II. Présentation de la plate-forme Oracle :

II.1 le serveur de données (Oracle data base) :

Oracle est un SGBD (système de gestion de bases de données) édité par la société du même nom (Oracle Corporation - <http://www.oracle.com/>), leader mondial des bases de données.

La société *Oracle Corporation* a été créée en 1977 par Lawrence Ellison, Bob Miner, et Ed Oates. Elle s'appelle alors *Relational Software Incorporated (RSI)* et commercialise un Système de Gestion de Bases de données relationnelles (SGBDR ou RDBMS pour *Relational Database Management System*) nommé *Oracle*.

En 1979, le premier prototype (RDBMS - RSI1) intégrant la séparation des espaces d'adressage entre les programmes utilisateurs et le noyau Oracle est commercialisé. Cette version est entièrement développée en langage assembleur. La seconde version (RDBMS - RSI2) est un portage de l'application sur d'autres plates-formes.

En 1983 la troisième version apporte des améliorations au niveau des performances et une meilleure prise en charge du SQL. Cette version est entièrement codée en langage C. A la même époque RSI change de raison sociale et devient *Oracle*.

En 1984 la première version d'Oracle (Oracle 4) est commercialisée sur les machines IBM.

En 1985 Oracle 5 permet une utilisation client-serveur grâce au middleware *SQL*Net*.

En 1986 Oracle a été porté sur la plateforme 8086.

En 1988 Oracle 6 est disponible sur un grand nombre de plates-formes et apporte de nombreuses nouvelles fonctionnalités ainsi qu'une amélioration notable des performances.

En 1991, Oracle 6.1 propose une option *Parallel Server* (dans un premier temps sur la DEC VAX, puis rapidement sur de nombreuses autres plates-formes).

En 1992, Oracle 7 sort sur les plates-formes UNIX (elle ne sortira sur les plates-formes Windows qu'à partir de 1995). Cette version permet une meilleure gestion de la mémoire, du CPU et des entrées-sorties. La base de données est accompagnée d'outils d'administration (*SQL*DBA*) permettant une exploitation plus aisée de la base.

En 1997, la version Oracle 7.3 (baptisée *Oracle Universal Server*) apparaît, suivie de la version 8 offrant des capacités objet à la base de données, et sont dans la version 8.1.5.0.0 Oracle a intégré les services Web. Des composants java web services ont été intégrés par native Sun.

En générale la version Oracle 9i est un sur ensemble d'Oracle 8i et contient des fonctions supplémentaire, l'architecture de base est toujours la même, le principale changement est l'augmentation en permanence, en nombre de paramètre modifiable rapidement sans redémarrer la base de données, l'administrateur peut modifier (en cour d'exploitation) le shared pool et le paramètre **DB_CACHE** (ancien blocks buffers :gestion dynamique de la mémoire), pour cela, il existe quelque nouveau paramètre dans le fichiers init.ora.

Oracle est écrit en langage C et est disponible sur de nombreuses plates-formes matérielles (plus d'une centaine) dont :

- AIX (IBM)
- Solaris (Sun)
- HP/UX (Hewlett Packard)
- Windows NT (Microsoft)

Oracle depuis la version 8.0.5 est disponible sous Linux

Les versions d'Oracle

Oracle se décline en plusieurs versions

- Oracle Server **Standard**, une version comprenant les outils les plus courants de la solution Oracle. Il ne s'agit pas pour autant d'une version bridée...
- Oracle Server **Enterprise Edition**

Les fonctionnalités d'Oracle

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents

IL2 le serveur d'application Oracle 9iAS (Internet Application Server) :

Le troisième serveur d'application le plus vendu, avec Oracle 9iAS c'est un serveur Apache qui est fourni. Ce dernier est le serveur HTTP le plus répandu de la planète (50% de part du marché). Apache se configure par modifications de fichiers textes. Ce type de configuration reste le plus puissant et on peut donc dire qu'Apache est le plus configurable des serveurs HTTP.

IL2.1 Oracle 9i Forms developer :

Oracle9i Forms Developer fait partie de la suite Oracle9i Developer Suite. Cette suite comprend entre autre les applications JDeveloper, Report Developer. Il s'agit d'un environnement de développement d'applications permettant aux développeurs de construire facilement et rapidement des formulaires complexes de bases de données. Il permet aux développeurs de construire des applications Java, optimisées pour Internet, sans écrire une seule ligne de code. En effet, cet environnement inclut un certain nombre d'assistants, d'utilitaires et de contrôles (cases à cocher, zones de texte, ...) pour améliorer la vitesse de développement de l'application. Il est aussi facile d'étendre une application Forms

en intégrant les propres composants Java du développeur, grâce aux JavaBeans et au Pluggable Java Components (PJC) de JDeveloper9i. En effet, JDeveloper9i, produit intégré dans la Suite Oracle9i Developer, inclut l'assistant Pluggable Java Components qui permet d'accéder à toutes les classes des composants Forms. Il suffit alors de choisir la classe d'un composant Forms à personnaliser, et JDeveloper génère un code squelette que le développeur peut ensuite modifier ou enrichir. Pour utiliser pleinement Oracle 9i Forms, deux environnements sont nécessaires : l'environnement de développement Oracle 9i Forms Developer et l'environnement de déploiement Oracle 9iAS Forms Service.

II.2.2 Oracle 9iAS Forms services:

- **Présentation générale :**

Oracle9iAS Forms Services fait parti du serveur d'applications Oracle9iAS (Application Server). Il regroupe un ensemble de composants Forms, utilisés pour déployer des applications Forms sur Internet. Il s'assure ainsi que les applications Oracle9iAS Forms peuvent être exécutées automatiquement et sur n'importe quel réseau.

- **Architecture de Oracle9i Forms Services :**

Oracle9iAS Forms Services est déployé selon une architecture à trois niveaux ('three tier architecture') :

- le niveau client, 'client tier',
- le niveau application, 'application tier' ou 'middle tier',
- le niveau base de données, 'database tier'

Oracle9iAS Forms Services étant intégré à Oracle9iAS, il sera installé au niveau du middle tier, sur le serveur d'applications.

Le niveau client - client tier :

Il contient le navigateur web où l'application est affichée.

Le client Forms est une applet Java générique qui s'exécute dans le navigateur de l'utilisateur. Il n'y a donc besoin de la télécharger qu'une seule fois et plusieurs applications Forms peuvent ensuite utiliser cette même applet Java.

Toute personne ayant un navigateur compatible Java peut exécuter une application Forms, sur n'importe quel réseau : Internet, Intranet ou Extranet.

Aucun traitement n'est exécuté sur le client, seulement l'affichage à l'écran. Tout le traitement est réalisé sur le niveau application (application tier) : le client reçoit un message du serveur, sur quoi afficher, et renvoie les actions réalisées par l'utilisateur au serveur.

Le niveau application - application tier/middle tier :

Il contient toute la logique de l'application.

Le client Forms communique avec le processus d'exécution Forms (Forms Runtime Process), par l'intermédiaire du listener HTTP, intégré dans Oracle9iAS.

Par défaut, une communication HTTP est utilisée, mais il est possible de configurer

Oracle9iAS pour utiliser SSL (HTTPS) afin d'avoir une communication sécurisée.

Le niveau base de données-database tier :

Il contient la base de données, lieu de stockage de toutes les données de l'entreprise. Cette base de données est normalement installée sur une machine dédiée ou sur un cluster de machines.

Voici un schéma regroupant l'architecture de oracle9i Forms Services et ses composants :

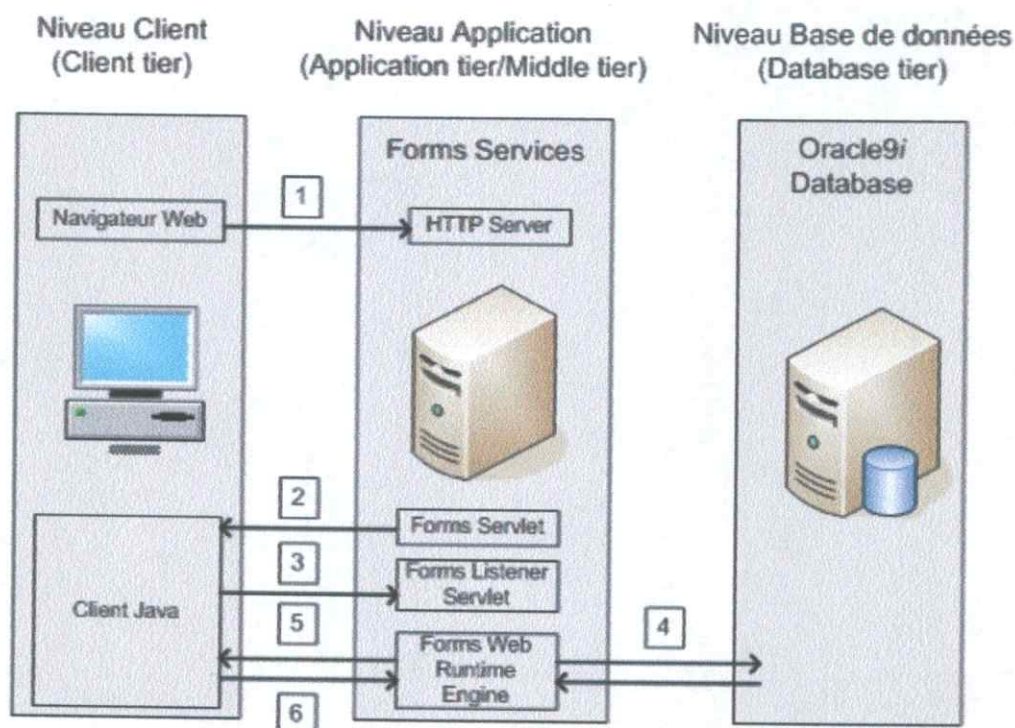


Figure V.2 Architecture d'Oracle 9i FORMS

IL2.3 Aperçu du processus d'exécution d'une application Oracle 9iForms :

Voici les étapes du processus d'exécution, lorsqu'un utilisateur veut utiliser une application Forms. Elles se basent sur le schéma indiqué plus haut.

Etape 1 - Entrée d'une URL :

L'utilisateur souhaitant utiliser une application Forms doit entrer l'URL de cette application dans le champ adresse de son navigateur. Cette URL pointe vers le serveur HTTP Oracle (Oracle HTTP Server - OHS), intégré dans Oracle9iAS, accédant ensuite au servlet Forms (Forms Servlet). Pour pouvoir exécuter correctement une application Forms, ce navigateur devra être compatible Java.

Etape 2 - Le servlet Forms (Forms Servlet)

Le servlet Forms est un composant de Forms Services. Il détecte automatiquement le type de

navigateur de l'utilisateur : Internet Explorer ou Netscape. Il retourne ensuite au client, un page HTML contenant des paramètres de démarrage de l'application. Ces paramètres sont lus à partir d'un template HTML et d'un fichier de configuration de l'application, situés sur le serveur.

Une fois récupérée par le client, cette page HTML initialise le téléchargement du client Java Forms et par la suite, permettra de démarrer l'applet Java (donc l'application), contenue dans le navigateur du client.

Lors du premier appel à l'application, les classes Java téléchargées depuis le serveur et utilisées côté client, sont mises en cache sur le navigateur de l'utilisateur, éliminant ainsi le besoin de re-télécharger les mêmes fichiers pour une utilisation future de l'application. Le client Java Forms sera de nouveau téléchargé seulement si une nouvelle version de l'application est détectée sur le serveur d'application (middle tier), assurant ainsi que le client utilise toujours la dernière version des classes Forms.

Etape 3 - Le Forms listener servlet :

Lorsqu'un utilisateur demande à démarrer une application Forms, le Forms Listener Servlet récupère la requête de connexion à l'application du client Java et initialise un processus d'exécution Forms (Forms Runtime Process).

La session de l'utilisateur est alors liée avec le Forms Runtime Process grâce à un cookie de session créé par le Forms Listener Servlet. Ce cookie est gardé en mémoire et attaché à l'URL de chaque demande du client et réponse du serveur.

Le Forms Listener Servlet prend aussi en charge :

- l'arrêt du Forms Runtime Process lorsque l'utilisateur ferme l'application Forms ou ferme la fenêtre du navigateur,
- les communications réseau entre le navigateur du client et le Forms Runtime Process associé.

Il n'est pas besoin de le configurer car il est déjà configuré pendant le processus d'installation de Oracle9iAS.

Etape 4 - le Forms Runtime Process :

Le Forms Runtime Process est donc créé quand un utilisateur veut accéder à une application Forms. Il a pour tâche de :

- gérer la logique et le traitement d'une application Forms (en fait exécuter le code de l'application),
- maintenir la connexion à la base de données.
Il joue 2 rôles :
- Quand il communique avec le navigateur du client, il agit comme un serveur en gérant les requêtes du navigateur du client et envoie les meta-données (description du contenu/de la structure des données) au client pour décrire l'interface de l'utilisateur
- Quand il communique avec le serveur de base de données, il agit comme un client en interrogeant la base pour demander des données.

- Les données utilisées par une application web Forms ne sont ainsi pas directement chargées sur le navigateur du client mais échangées entre le serveur d'applications et le serveur de base de données. Cela permet donc une amélioration des performances et une réduction du trafic réseau.

Etape 5 - Communication vers le client Java Forms :

Le Forms Web Runtime communique avec le client Java en utilisant le protocole HTTP ou HTTPS. Au lieu d'envoyer le texte entier pour chaque message, il compare le contenu de chaque message avec le message envoyé précédemment, et ne retient ensuite que les différences entre les deux. C'est cette différence, ensuite, qui est envoyée sur le réseau vers le client.

Etape 6 - le client Java Forms :

Le client Java Forms fournit l'interface utilisateur.

Il consiste en un ensemble de classes génériques Java qui sont téléchargées sur le client depuis le serveur Oracle9iAS, la première fois qu'un utilisateur veut accéder à une application Forms.

C'est une applet 100% java qui est exécutée par la JVM (Java virtual Machine) du navigateur ; il est donc nécessaire d'avoir une JVM installée sur la machine cliente.

Cette JVM peut être le plug-in Java de Sun, Oracle Jinitiator ou la VM native Microsoft Internet Explorer. La même applet java est utilisée pour différentes applications Forms. Elle est donc téléchargée une seule fois et ensuite mise en cache sur le client. Elle sera ainsi disponible pour toutes les applications Forms suivantes.

Le Forms Java client utilise un mécanisme 'intelligent' pour communiquer avec le serveur, réduisant le trafic réseau au minimum.

II.2.4. Oracle Jinitiator :

Sur les plateformes Windows, Oracle fournit Jinitiator, qui a des avantages sur d'autres JVM lorsque des applications Forms s'exécutent sur Internet. Oracle Jinitiator est la version Oracle du plug-in Java de Sun. Il démarre comme un Plug-in pour Netscape et comme un composant ActiveX pour Internet Explorer, permettant aux utilisateurs d'exécuter des applications oracle9iForms depuis Netscape ou Internet Explorer. Voici les principaux avantages de Jinitiator :

- permet à une JVM récente et certifiée Oracle, de s'exécuter dans des anciennes versions de navigateur,
- fournit des extensions à une JVM de base, tel que le HTTPS ou le SSL,
- c'est un environnement de déploiement. Les fichiers de classe d'applications sont ainsi automatiquement mis en cache, fournissant un démarrage plus rapide des applications.
- il s'installe de lui-même et s'auto update, comme un plug-in ou un contrôle ActiveX.

II.3 le poste client : c'est un simple navigateur web : Netscape ou Internet Explorer.

III. Réalisation et mise en œuvre de l'application :

III.1 Etapes de réalisations :

Cette étape consiste à mettre en place les meilleures conditions pour l'environnement destiné à l'utilisateur de l'application développée. Les différentes activités de cette étape sont :

- Installation du serveur de données Oracle 9i.
- Définir un nouveau utilisateur oracle.
- Définir le schéma base de donnée : les tables, les contraintes, les vues,...
- Installer l'outil Oracle 9i Developer Suite : inclut le serveur d'application Oracle 9iAS et l'outil de développement Oracle Forms 9i.
- Développer l'application sous Oracle Forms 9i.
- Sécuriser l'application en attribuant des mots de passe et des privilèges aux utilisateurs
- Configurer Oracle 9iAS pour déployer l'application

III.2 Présentation de l'application :

1. Authentification :

La première étape consiste à démarrer un navigateur web et saisir la chaîne de connexion :

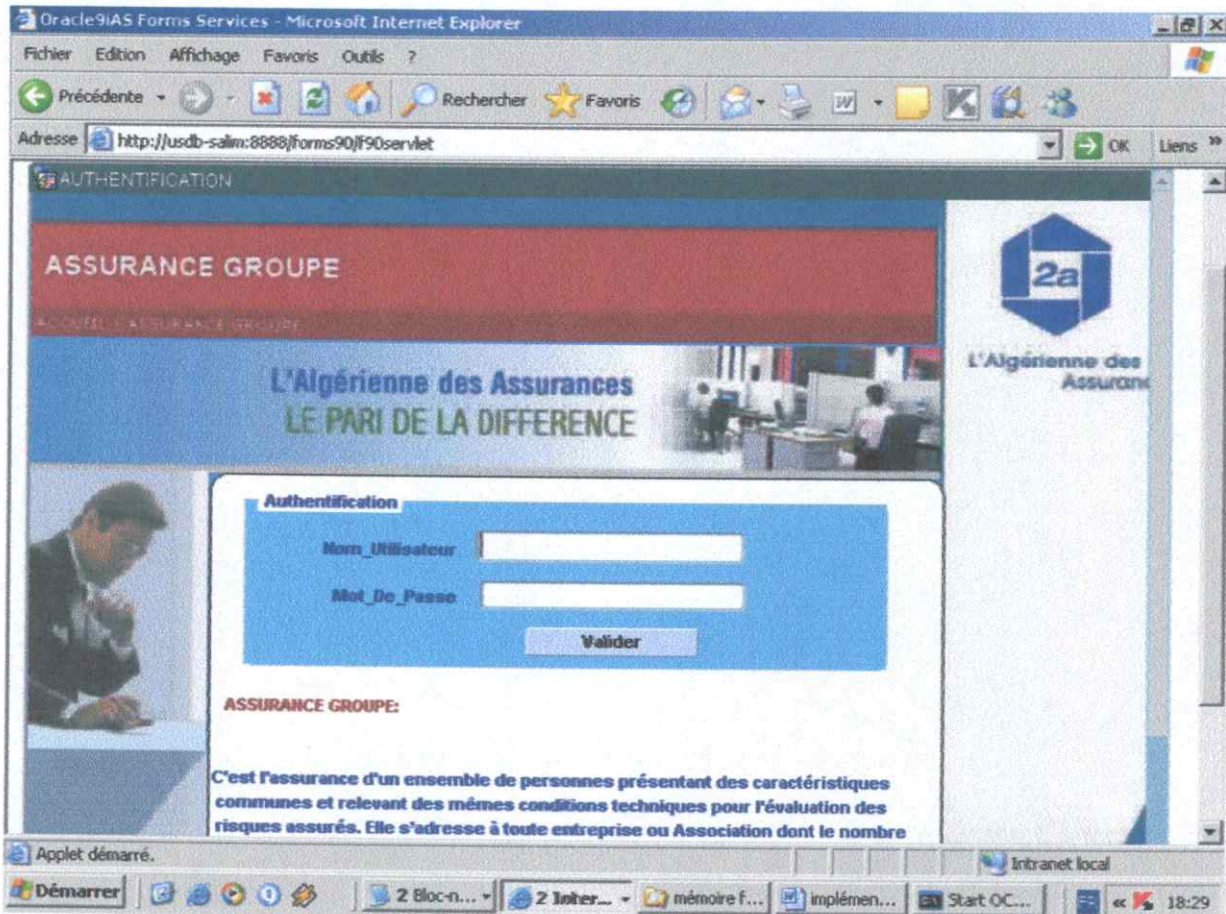


Figure V.3. Authentification

Cette première fenêtre consiste à l'authentification, l'application est sécurisée, c'est-à-dire l'accès est autorisé qu'en fournissant un nom utilisateur et un mot de passe. Dans cette application l'utilisateur soit un administrateur, soit un producteur d'agence.

2. Fenêtre de producteur d'agence :



Figure V.4 fenêtre principale de producteur

Cette fenêtre doit apparaître si l'utilisateur de l'application est un producteur d'agence, il doit sélectionner une opération :

- Production : déclaration des nouvelles polices, avenant, clients ainsi la suppression.
- Sinistre : déclaration des nouveaux sinistres, remboursements
- Consultation : consulter les polices, les avenants, les clients, les sinistres

2.1 Production :

2.1.1 Police :

En cas l'utilisateur choix l'opération production, la fenêtre suivante doit apparaître :

L'agence de Producteur

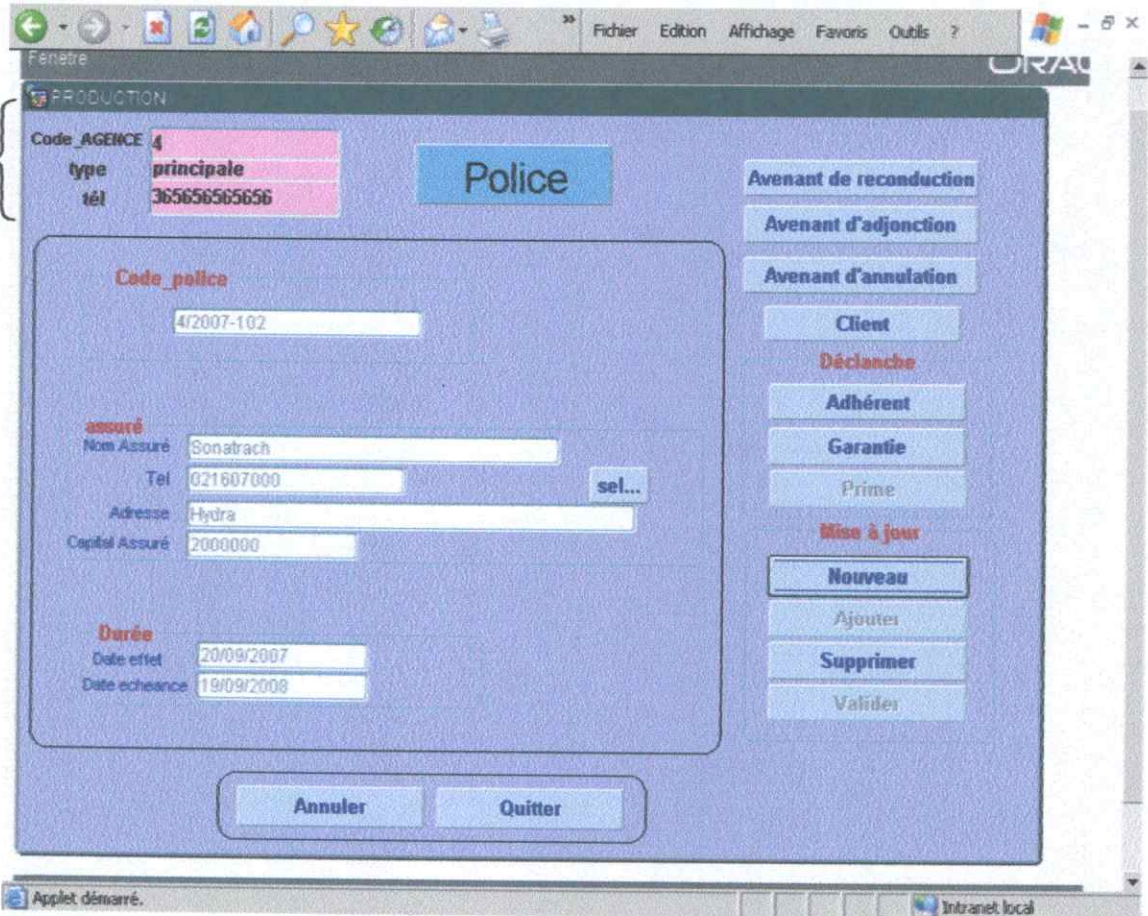


Figure V.5 Police

Cette fenêtre permet la création des nouveaux police ainsi ses adhérents, ses garanties et de calculer la prime à payer. En plus, le producteur peut déclarer des avenants (reconduction, annulation, d'adjonction).

2.1.2. Adhérent :

The screenshot shows a web application window titled 'Fenêtre' with the application name 'ORAC'. The main content area is titled 'Adhérent' and contains a form for entering member details. The form fields are as follows:

Field	Value
Code adhérent	4/2007-102
Code police	1
Nom	Dechir
Prenom	Mohamed
Date de naissance	26/08/1970
Age	38
Adresse	cité gare,hadjout, tipaza
Num_Sécurité Social	1236555656655
salaire	30000
DA	
Situation	<input checked="" type="radio"/> Célibataire <input type="radio"/> Marié

On the right side of the form, there is a 'Rechercher' section with a 'Recherche' button. Below it is a 'mise à jour' section with buttons for 'nouveau', 'ajouter', 'supprimer', and 'valider'. At the bottom of the form, there are 'Annuler' and 'Quitter' buttons. The window title bar shows 'Fenêtre' and the application name 'ORAC'. The status bar at the bottom indicates 'Applet démarré.' and 'Intranet local'.

Figure V.6 Adhérent

Cette fenêtre permet la déclaration des adhérents pour une police déjà déclarée ainsi ses enfants et ses conjoints en cas de situation marié. De plus, le producteur peut supprimer des adhérents en cas d'avenant d'annulation.

2.1.3 Garanties :

Listes des garanties

Police	Code	Libélé
4/2007-103	g02	Décès par accident
4/2007-103	g1	Décès/AD
4/2007-103	g3	Décès par accid de circulation
4/2007-103	g7	ipp
4/2007-103		

Supprimer Ajouter Fermer

Figure V.7 Garantie

Cette fenêtre permet l'insertion des garanties pour une police déjà déclarée ou supprimer une garantie en cas d'insérer deux fois la même garantie.

2.1.4 Prime :

Oracle9iAS Forms Services - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Rechercher Favoris

Adresse <http://usdb-salim:8888/forms90/f90servlet> OK Liens »

Prime

L'Algérienne des Assurances

Les garanties

- Garantie de base(Décès/IAD)
- Décès par accident
- Rente Education
- IPP
- décès par accident de circulaion
- Maladie

Option1
Option2

Prime

Prime_net 38812,582

Droit timbre 18

Accessoire 100

Prime_totale 38922,542

Select

Reduction

Valider

Applet démarré. Intranet local

Démarrer Start O... lienhttp... Oracle... impléme... Sans tit... 20:05

Figure V.8 Prime

Cette fenêtre permet le calcul de prime à payer en sélectionnant les garanties nécessaire. Le bouton 'Réduction' permet la négociation de la prime en réduisant la prime totale.

2.1.5 Avenant :

Oracle9IAS Forms Services - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Rechercher Favoris

Adresse <http://usdb-salmc:8888/forms90/f90oservlet> OK Liens »

PRODUCTION

Code 13

Avenant

Code Avenant 4/2007-110 Code Police 4/2007-104 sel...

Nom Assuré Sonatrach

Adresse Hydra

Tel 021607000

Date Effet 20/09/2007 Date Echeance 19/09/2008

Droit Timbre 10 sel...

Accessoire 300 Prime Total 300

Annuler Fermer

2a L'Algérienne des Assurances

Declanche

Adhérent

Garantie

Mise à jour

Nouveau

Eregistrer

Applet démarré. Intranet local

Démarrer Start O... lienhttp... Oracle... impléme... Sans tit... 20:16

Figure V.9 Avenant

Cette fenêtre permet la déclaration d'un type avenant selon le code affiché en dessus d'image, les bouton 'Adhérent' et 'Garantie' ont une importance en cas d'avenant d'annulation pour la suppression ou avenant d'adjonction pour l'insertion.

2.2 Service Sinistre :

2.2.1 : Déclarer un sinistre :

Cette fenêtre permet la déclaration d'un sinistre pour une police déjà déclaré

The screenshot shows a web browser window titled "Oracle9iAS Forms Services - Microsoft Internet Explorer". The address bar displays "http://usdb-salin:8888/forms90/f90servlet". The main content area is a form titled "FENETRE1" with a header "Déclarer Sinistre". On the left, there is a sidebar with the following information: "Agence: 4", "Type: principale", and "Téléphone: 3656565656". The main form contains the following fields: "Code sinistre" (4/2007-27), "Code Police" (4/2007-103), and "Date Sinistre" (21/09/2007). Below these fields are three buttons: "Annuler", "Ajouter", and "Nouveau". At the bottom center, there is a "Quitter" button. The Windows taskbar at the bottom shows the system tray with the time 17:36 and several icons, including "Démarrer", "Implémentation - M...", "Start OC4J Instance", and "Oracle9iAS Form...".

Figure V.10 Sinistre

2.2.2 Remboursement :

Cette fenêtre permet le règlement d'un sinistre déjà déclaré

The screenshot shows a web browser window titled "Oracle9iAS Forms Services - Microsoft Internet Explorer". The address bar displays "http://usdb-salin:8888/forms90/f90servlet". The main content area is a form titled "FENETRE1" with a header "Remboursement". On the left, there is a sidebar with the following information: "Agence: 4", "Type: principale", and "Téléphone: 3656565656". The main form contains the following fields: "Code_dossier" (4/2007-21), "Code_Sinistre" (4/2007-26), "Code_Police" (4/2007-4), "Adhérent Numéro" (4), "Nom" (dsf), "Prenom" (fs), "Garantie" (21), "Date" (21/09/2007), and "Montant" (5000 DA). Below these fields are three buttons: "Annuler", "Ajouter", and "Nouveau". At the bottom center, there is a "Quitter" button. The Windows taskbar at the bottom shows the system tray with the time 17:40 and several icons, including "Démarrer", "Implémentation - M...", "Start OC4J Instance", and "Oracle9iAS Form...".

Figure V.11 Remboursement

Note : la suppression d'une police déclenche automatiquement la suppression des avenants, des adhérents, des garanties, des sinistres et les remboursements de cette police.

2.3 Consultation : si le producteur d'agence sélectionne l'opération 'Consultation' la fenêtre suivante doit apparaître :

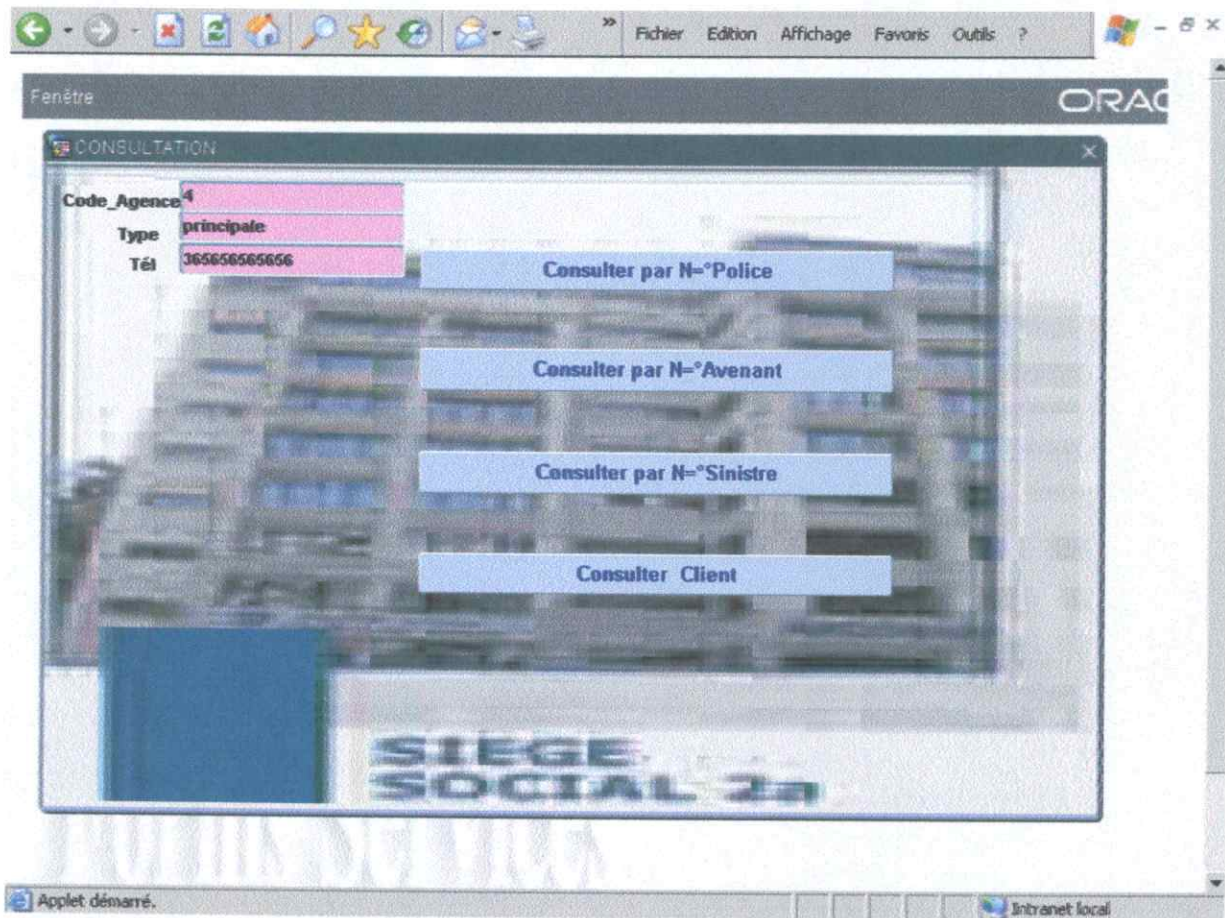


Figure V.12 Consultation

Le producteur doit choisir une consultation parmi les consultations ci-dessus.

2.3.1 Consultation par N°police :

Le producteur sélectionne un code police à rechercher, tous les renseignements de la police sera afficher tel que :

- Les clients
- Les adhérents
- Les garanties
- Les avenants de la police

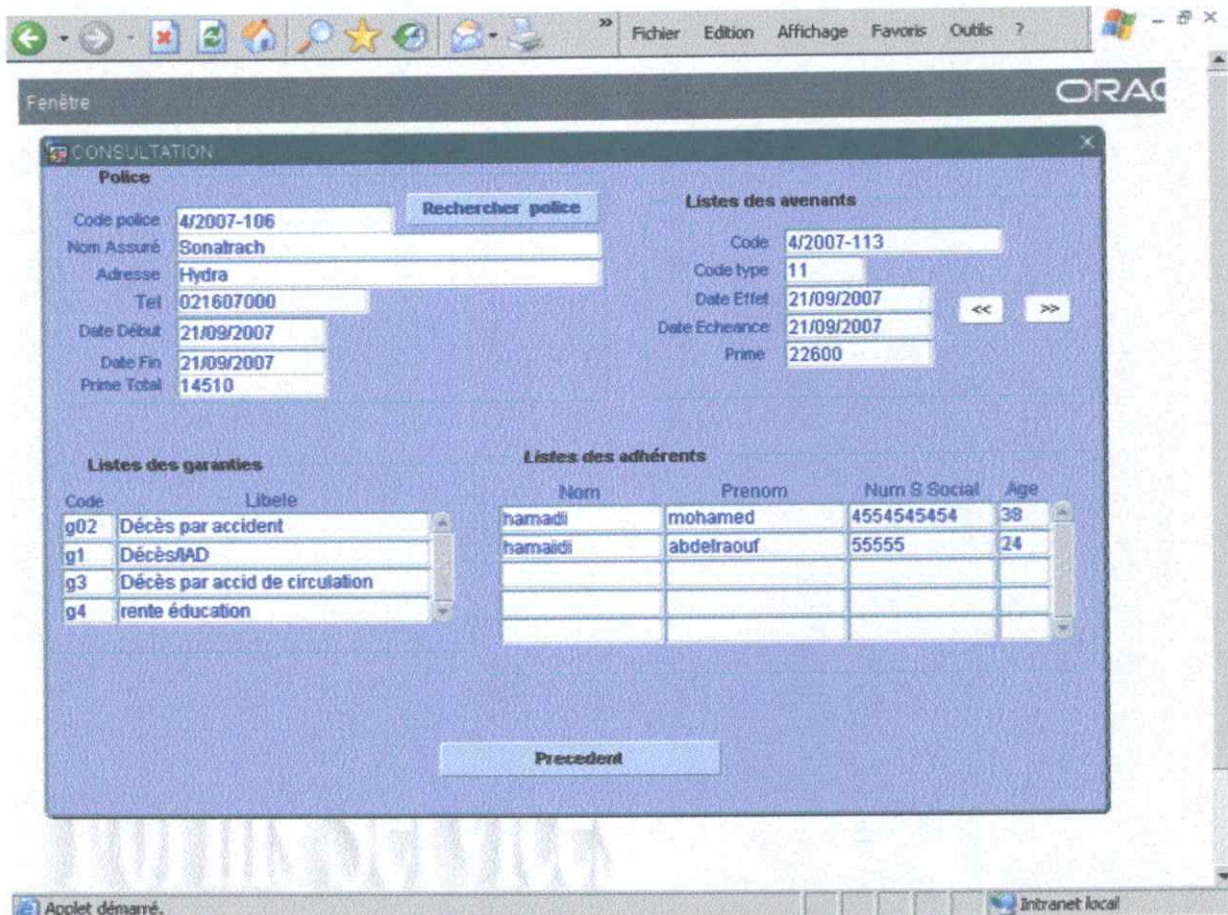


Figure V.13 Consultation police

2.3.2 Consultation par N^o Avenant :

Le producteur sélectionne un code avenant à rechercher, tous les renseignements de l'avenant sera afficher tel que :

- La durée
- Le type
- La police concernée

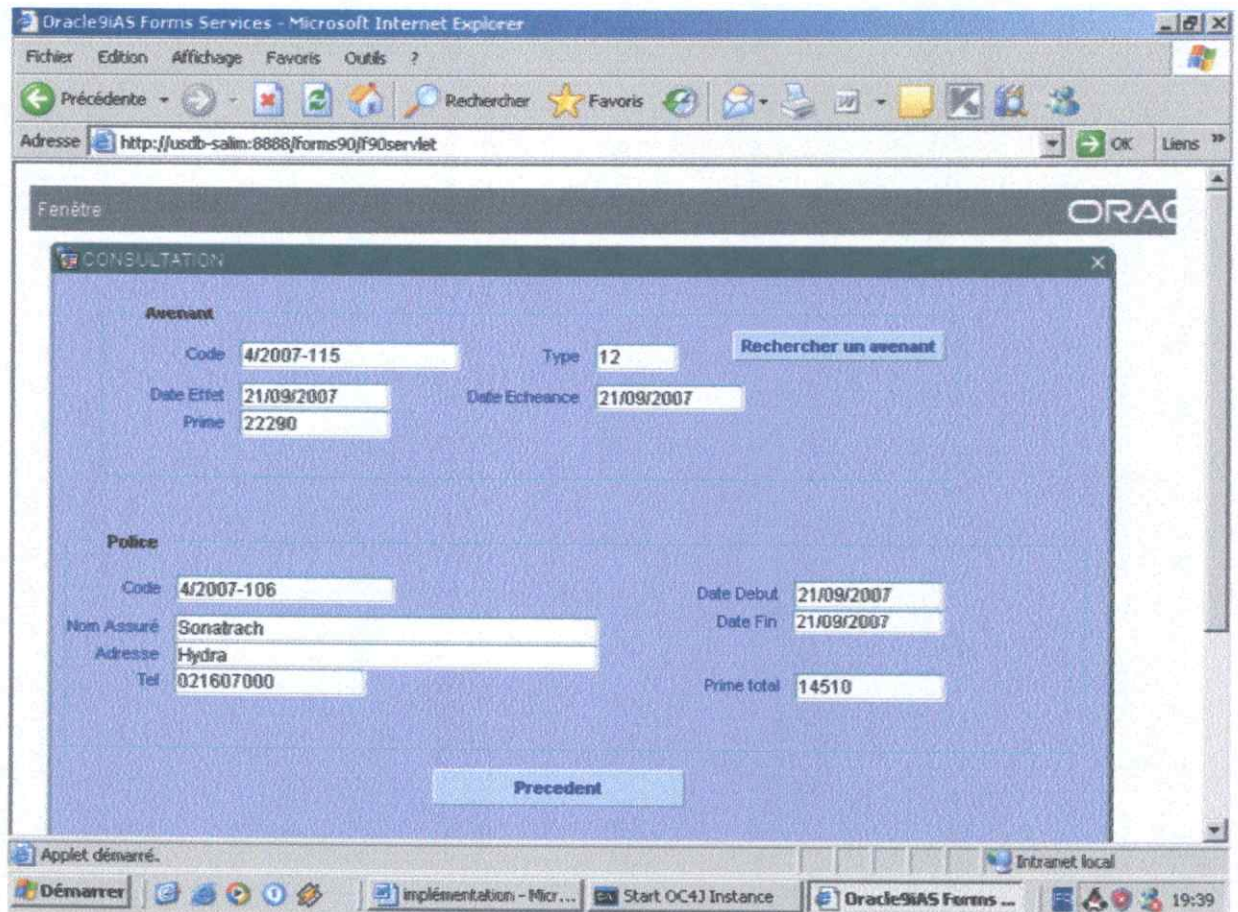


Figure V.14 Consultation Avenant

2.3.3 Consultation par N^o Sinistre:

Le producteur sélectionne un code sinistre à rechercher, tous les renseignements de sinistre sera afficher tel que :

- Police concernée
- Date sinistre
- Remboursements déclarés pour ce sinistre

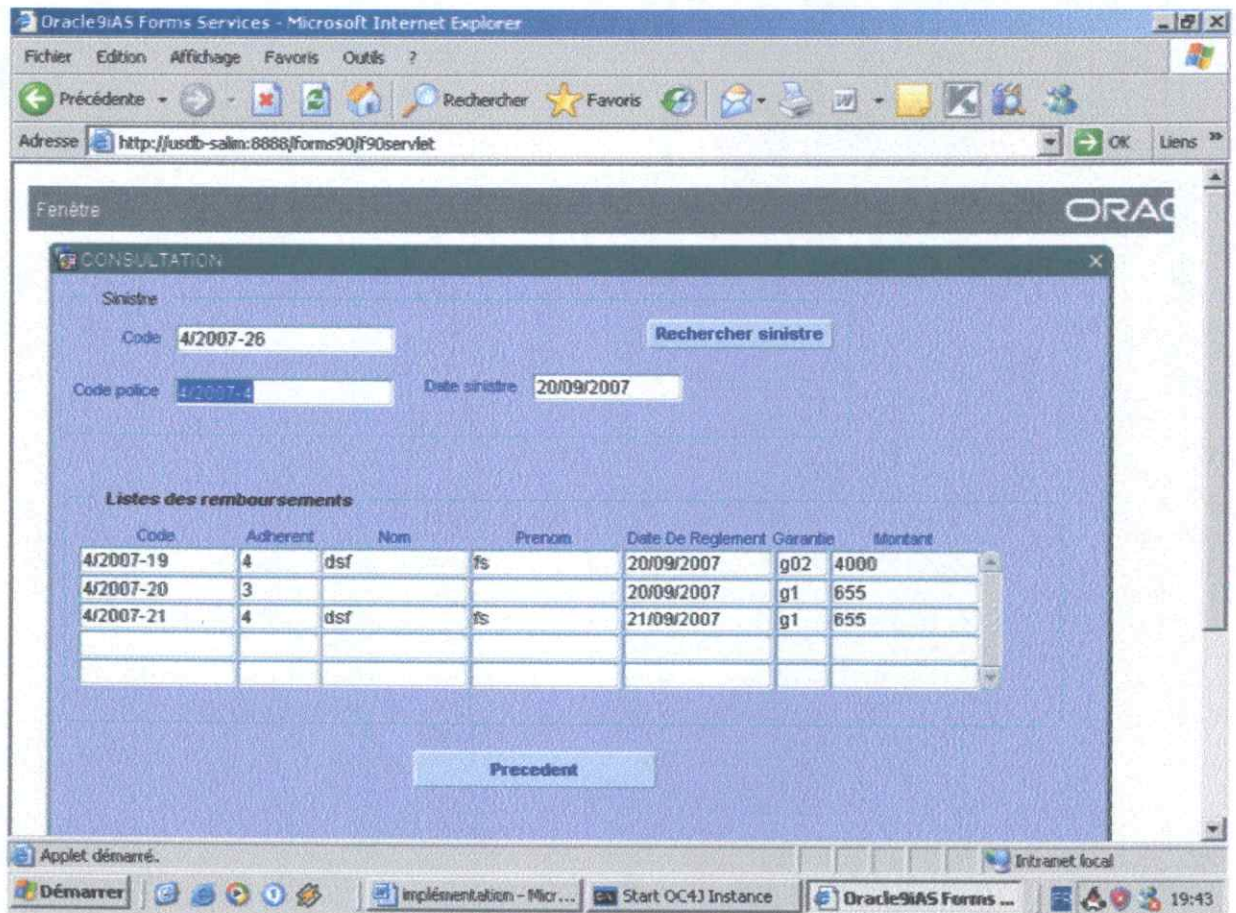


Figure V.15 Consultation Sinistre

2.3.4 **Consultation par nom client :** Le producteur sélectionne un client à rechercher, tous les renseignements de client sera afficher.

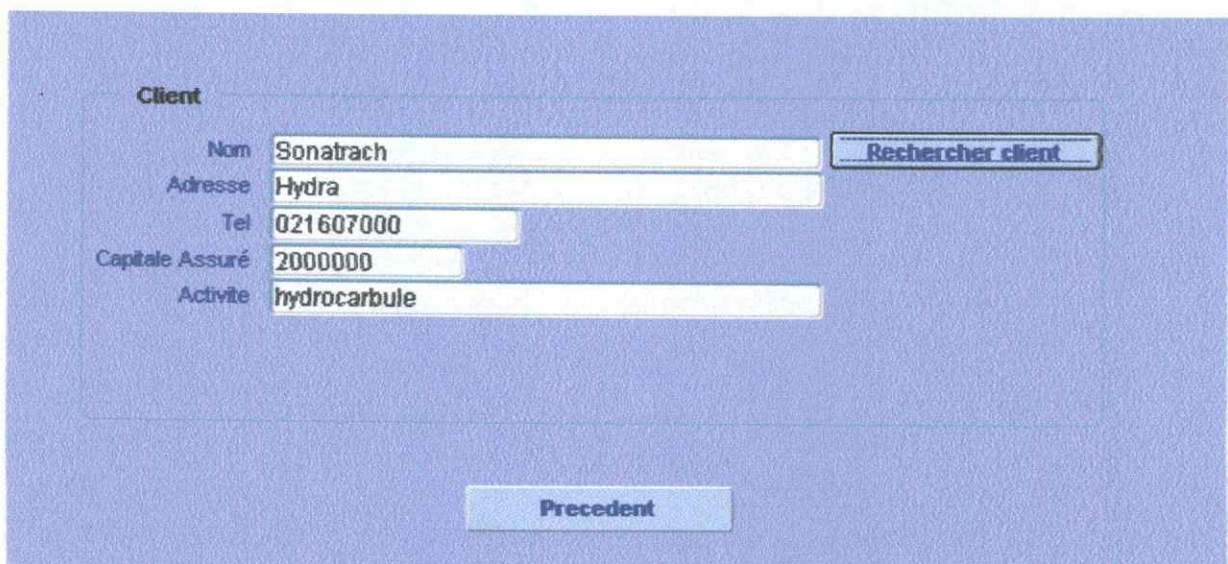


Figure V.16 Consultation Client

3.Fenêtre de l'administrateur :

L'administrateur est chargé à la gestion des utilisateurs, des délégations, des agences, des garanties.

1.1 Gestion des délégations :

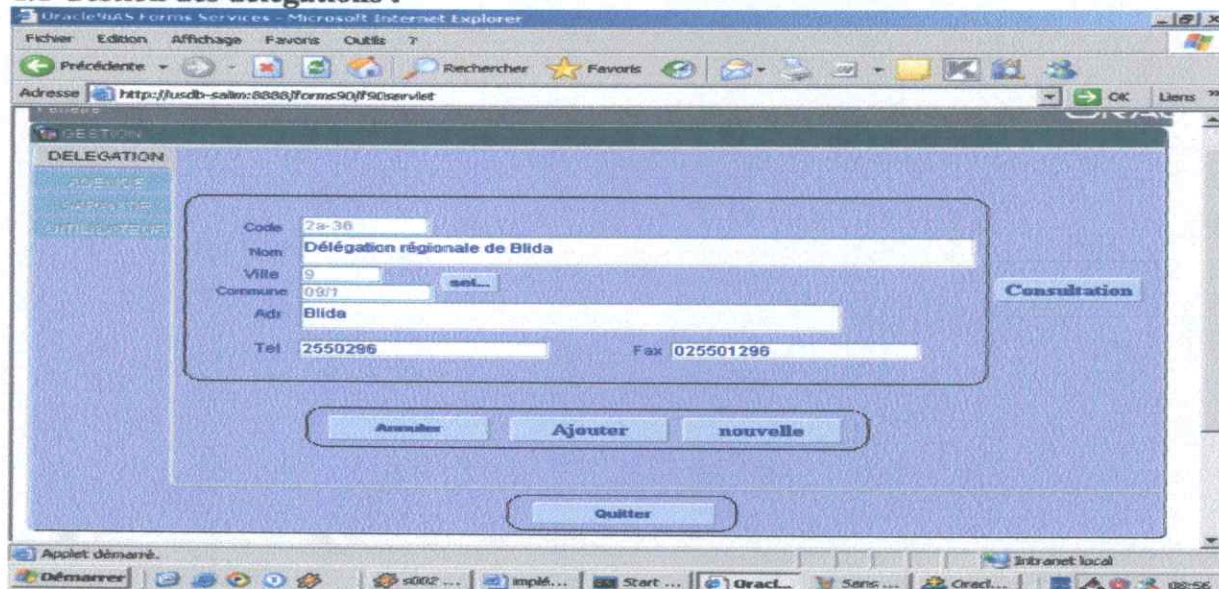


Figure V.17 délégation

2.11 Consultation :

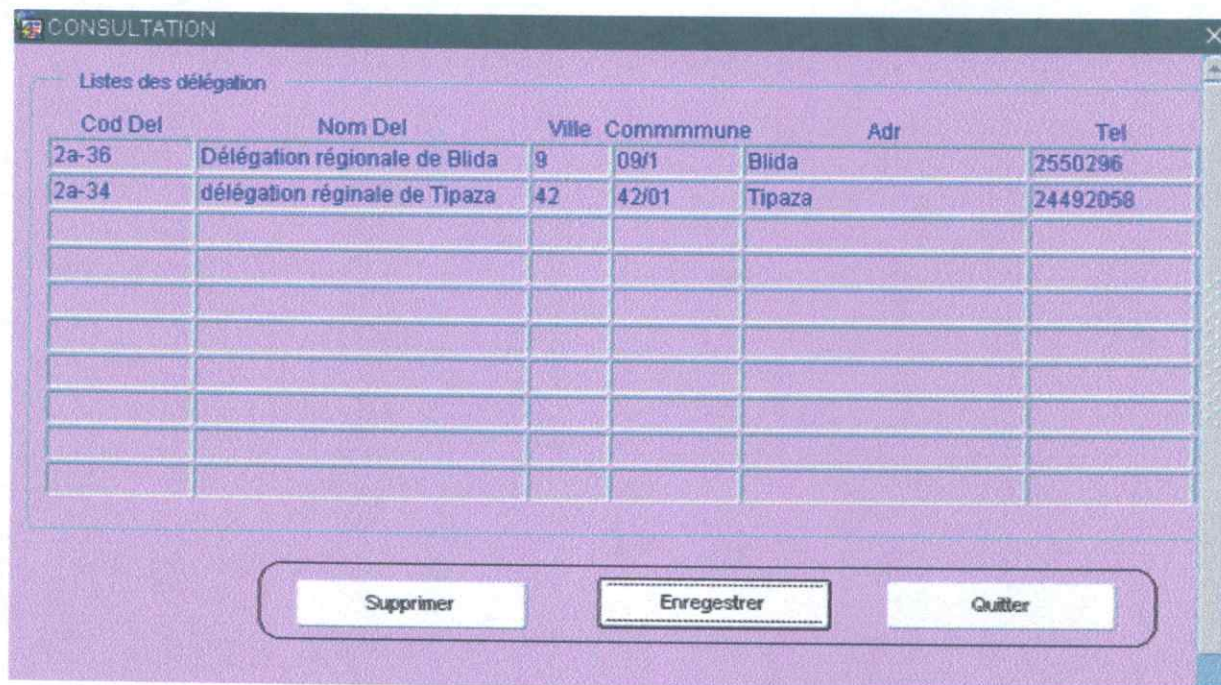


Figure V.18 Consultation délégation

2.2 Agence :

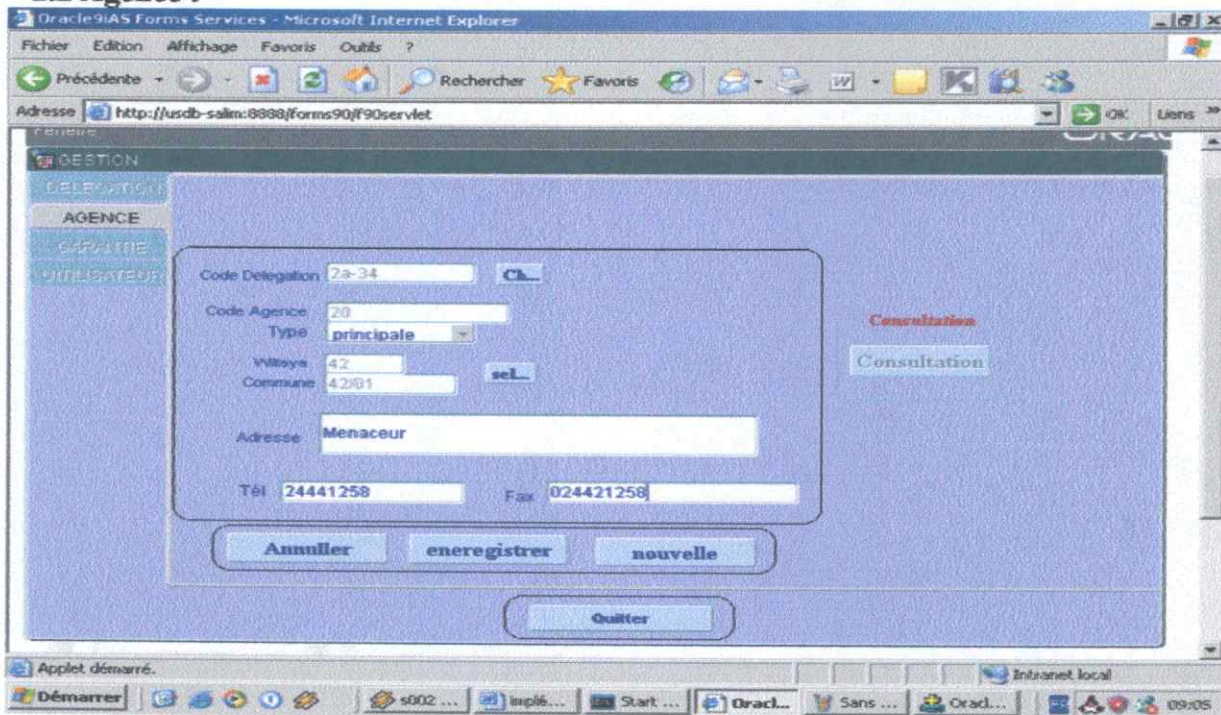


Figure V.19 Agence

2.3 Garanties :

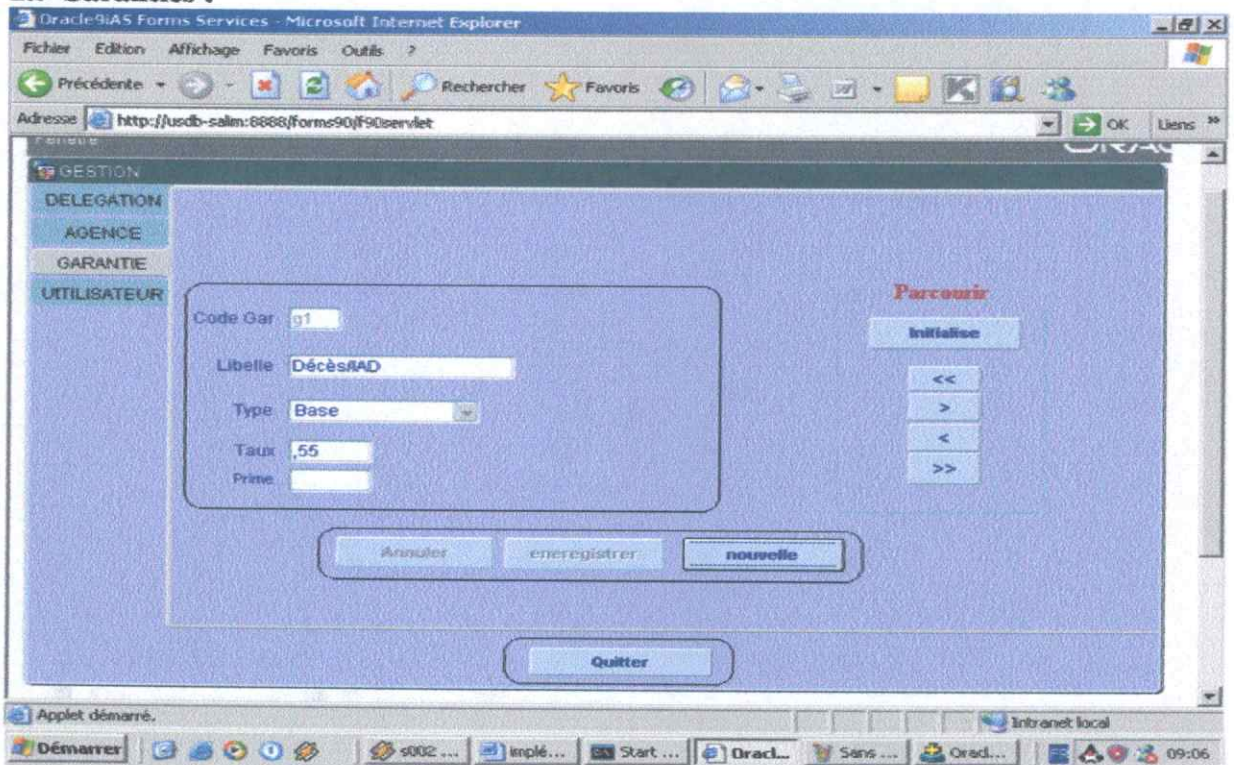


Figure V.20 Consultation Agence

2.4 Utilisateur (producteur d'agence) :

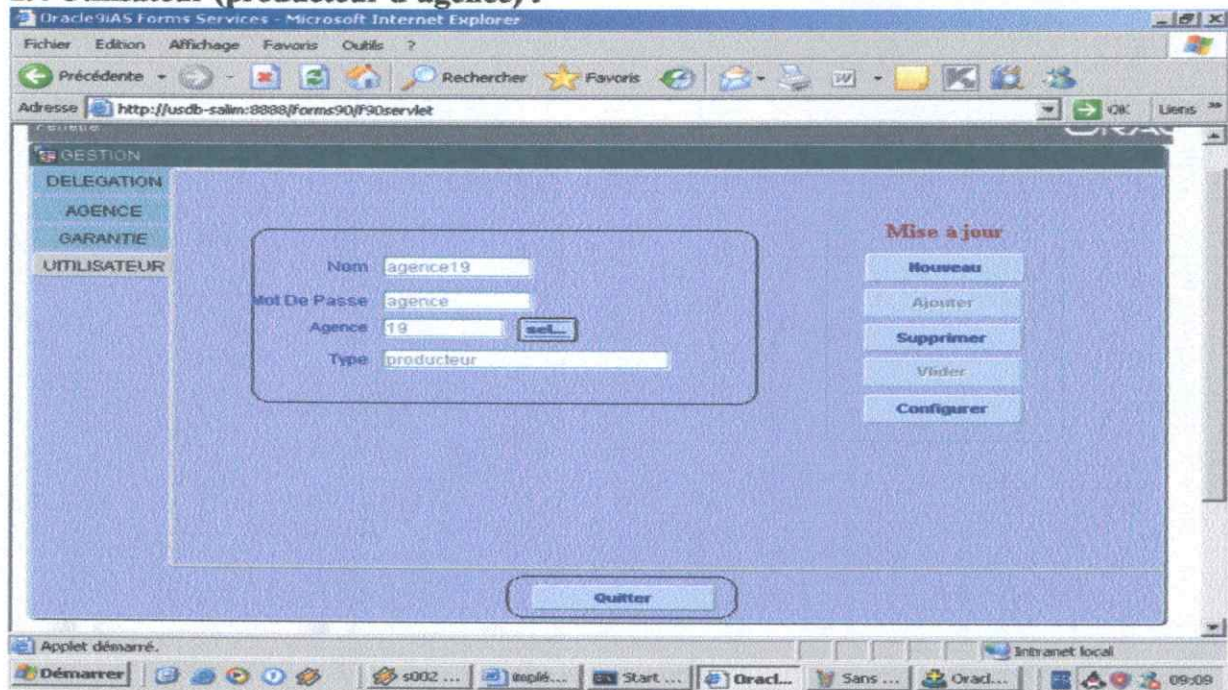


Figure V.21 Producteur

2.3.1 Configurer : permet la modification de mot de passe

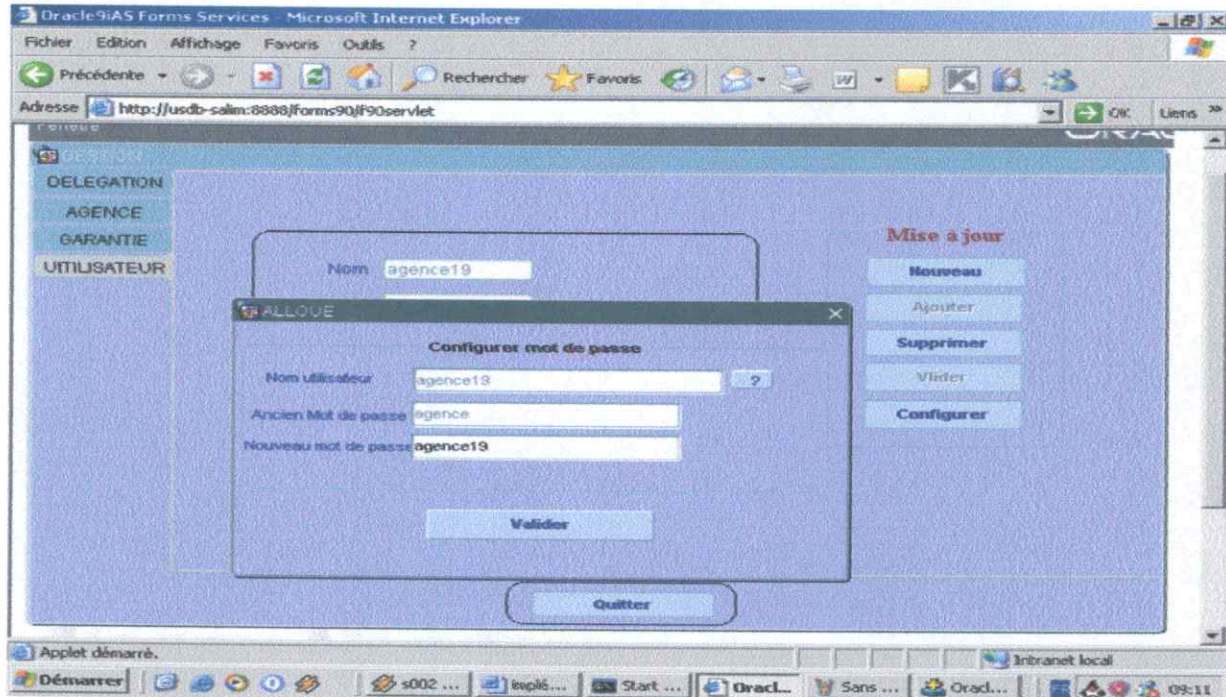


Figure V.21 Configurer mot de passe

IV. Conclusion :

Dans le présent chapitre nous avons présenté les principaux outils utilisés pour la réalisation de notre travaille ainsi une description générale de notre application.

Conclusion générale :

Notre travail était : la Conception d'un système d'information pour la commercialisation et gestion du sinistre du produit « **Assurance Groupe** », et la mise en place de la solution avec l'architecture 3 tiers. Nous avons opté à une plate-forme 100% oracle avec une conception UML.

Notre base de donnée est centralisée au niveau d'un serveur de donnée « Oracle 9i » accessible via une application déployée au niveau d'un serveur d'application « Oracle9iAS », l'application est aussi accessible via un navigateur web « Internet Explorer, Netscape ».

Sur le plan personnel, ce stage nous a permis :

- ✓ L'expérience que nous avons vécue à l'algérienne des assurance 2a pendant toute la durée du stage nous a permis non seulement d'améliorer notre savoir-faire, mais aussi notre savoir être.
- ✓ Découvrir l'environnement oracle
- ✓ Installer une base oracle
- ✓ Configurer un serveur d'application oracle
- ✓ Utiliser le langage de modélisation UML
- ✓ Bien que ce stage a été effectué dans une entreprise, nous avons adopté la rigueur dans le développement en prenant soin de spécifier et concevoir avant de réaliser l'application.

Cela a nous permis de dérouler les différentes phases du cycle de vie du logiciel, de la phase d'analyse jusqu'à celle des tests.

Toute au long de cette étude, nous avons essayé d'être à l'écoute des employés charger du produit « Assurance Groupe » et d'opter l'application pour qu'elle répond au niveaux à leur besoins. Nous espérons que ce modeste travail leur soit bénéfique.

Bibliographie

Les ouvrages :

N=°	Thème	Année	Auteur
[01]	Le langage SQL et l'outil PL*SQL Volume1, volume2, volume3	1998	Oracle Corporation
[02]	Oracle developer forms5 Partie I, II, III	1998	Oracle Corporation
[03]	Les bases de données Oracle 8i	2001	Roger Chapuis
[04]	Oracle 8 :développement de bases de données	2000	W.DAVID LOCKMAN
[05]	Le guide de l'utilisateur UML	2001	BOOCH, RUMBACH ,JAMES ,JAC OBSON
[06]	UML2 Par la pratique	1993	Rocques Pascal

Les sites web :

www.assurance-2a.com
www.supinfo-project.com/fr 2004
www.commentçamarche.com
otn.oracle.com
www.develoopez.fr