

Stage de fin de cycle 2007

**« Automatisation du
contentieux de l'UAB-vie »**
(Aout - Octobre 2007)

SINARE Judith et TAPSOBA Abdoul Malick

Abréviations.....	3
Introduction	4
A. GENERALITES	5
I. Présentation de l’UAB-vie	5
I.1 Historique	5
I.2 Evolution	5
I.3 Situation actuelle.	5
I.4 Environnement et performances	6
I.5 Les Agences et les Produits de l’UAB-vie	7
a. Les Agences	7
b. Les Produits de l’UAB-vie	7
II. Domaine d’étude	9
II.1 Le recouvrement et contentieux	9
II.2 Problématique	9
II.3 Langage et méthode d’analyse	10
a. Langage de modélisation : UML	10
b. Méthode d’analyse : RUP	12
II.4 Les acteurs du projet	15
II.5 Planning du stage	15
B. ETUDE DE L’EXISTANT	15
I. Objectif	15
II. Démarche d’analyse	16
III. Modélisation du processus métier	16
III.1 Diagramme des cas d’utilisations	16
III.2 Diagramme de séquence	23
IV. Gestion des exigences	30
IV.1 Besoins fonctionnels	30
a. Diagramme de communication	31
b. Recueil des informations	33
IV.2 Besoins techniques	35
a. Architecture logicielle	35
b. Outils de développement	37
V. Diagnostic du système existant	38
C. SCENARII	38
I. Objectifs	38
II. Scénario d’exception	38
III. Scénario de compromis	44
IV. Scénario idéal	47
V. Comparaisons des scénarii	50
D. SCENARIO RETENU	52
E. CONCEPTION DETAILLEE DE LA SOLUTION RETENUE	52
I. Reconfiguration du système	52
II. Diagrammes	53
II.1 Diagramme de cas d’utilisation	53
II.2 Diagramme de séquence	65
II.3 Diagramme d’activités	69
II.4 Diagramme de classes	73
II.5 Diagramme d’état-transitions	87
III. Etude technique	93

III.1 Description du scénario retenu	93
III.2 Langages et Outils de développement	97
IV. Protection de l'application	98
IV.1 Procédures de secours	98
IV.2 Procédures de sécurité	99
IV.3 Procédures de mise en oeuvre	100
Conclusion.....	101
Annexes	102
Bibliographie.....	1036

Abréviations

RC : **R**ecouvrement et **C**ontentieux

SCV : **S**ervice **C**lientèle et après **V**ente

UML: **U**nified **M**odeling **L**anguage

RUP: **R**ational **U**nified **P**rocess

HTTP: **H**yper**T**ext **T**ransfert **P**rotocol

COCOMO: **C**Oⁿstructive **C**O^st **M**odel

Introduction

L'informatisation de certaines tâches de l'entreprise est une étape nécessaire pour toute entreprise voulant faire face à la concurrence et améliorer la qualité de ses prestations. L'Union des Assurances du Burkina-vie (UAB-vie) ne déroge pas à la règle et s'inscrit dans ce processus d'informatisation de ses activités. Ainsi, nous avons durant trois (03) mois travaillé sur le thème « Automatisation du contentieux de l'UAB-vie ». Le contentieux a pour objectif de traiter les réclamations des assurés qui cotisent mensuellement sur leurs contrats afin de prendre position sur la véracité desdites réclamations. Le traitement actuel du contentieux se fait manuellement et rend les décisions du contentieux difficiles et souvent erronées. Notre stage a consisté à l'automatisation du traitement des réclamations des assurés afin de rendre le traitement plus efficace et les décisions plus fiables.

Ce travail a été réalisé dans le cadre de notre formation à l'Ecole Supérieure d'Informatique (ESI) et constitue notre projet de fin d'études du Cycle des Ingénieurs en Travaux Informatiques (CITI). Le projet a pour objectif de rendre l'étudiant apte à travailler en milieu professionnel.

Ce document présente les différentes étapes de notre étude. Il présente tout d'abord l'UAB-vie et la problématique de notre étude, ensuite la solution apportée pour résoudre la problématique et enfin quelques interfaces du futur système.

A. GENERALITES

I. Présentation de l'UAB-vie

I.1 Historique

L'union des assurances du Burkina (UAB) a vu le jour le 8 novembre 1990 avec l'obtention de son agrément suivant le Raabo numéro AN/VIII 019/FP/DCA.

Cette société est le fruit de la volonté d'opérateurs économiques Burkinabés d'investir dans le secteur des assurances jusque là occupé au Burkina-Faso par deux sociétés d'assurances qui se partageaient l'ensemble du marché.

L'UAB tire son nom Union de par son appartenance au groupe Français UAP, Union des Assurances de Paris, aujourd'hui racheté par le groupe AXA corporate solution. Elle a démarré effectivement ses activités en Janvier 1991 avec la branche IARDT (Incendie, Automobile, Risques Divers et Transport). La branche vie de l'UAB suivra en août 1992.

Société anonyme, l'UAB-vie avait un capital initial de deux cent soixante dix millions (270.000.000) entièrement libéré à sa création et détenu à 80% par des privés Burkinabés et à 20% par l'union Africaine (UA) une entreprise d'assurance ivoirienne différente de l'Union Africaine institution continentale.

I.2 Evolution

Depuis l'an 2000, suivant la réglementation du code CIMA (Conférence Interafricaine des Marchés d'Assurances). L'UAB devient un groupe constitué de l'UAB-IARDT et de l'UAB-vie au capital de un milliard (1.000.000.000) de francs CFA entièrement libéré dont 90% des parts détenues par les privés nationaux et 10% par l'AXA Côte d'Ivoire (ex union Africaine).

- l'UAB-IARDT détient la moitié du capital soit 500.000.000 F CFA et gère 64,26% du portefeuille du groupe et occupe 15,66% du marché national des Assurances-IARDT
- l'UAB-Vie, qui a reçu son agrément en novembre 2001, détient la moitié restante du capital social, entièrement libéré soit 500.000.000 et gère à son tour 35,74% du portefeuille du groupe et 41,09% du marché national des assurances – Vie.

I.3 Situation actuelle

L'UAB-vie dispose d'un conseil d'administration investi des pouvoirs les plus étendus pour agir en toutes circonstances au nom de la société.

La direction générale de l'UAB-vie à son actif 20 agents. A cet effectif s'ajoute neuf (09) agences commerciales réparties dans le pays et qui sont des réseaux de commercialisation.

L'UAB-vie est située sur l'avenue Kwamé N'Krumah en face du Liptako Gourma.

Elle a pour activité principale la production et la vente de contrats en assurance-vie.

Elle a d'autres activités accessoires comme le placement de ses fonds dans les institutions financières, l'investissement dans l'immobilier.

I.4 Environnement et performances

a. Environnement

- **Le contexte économique et social**

Le Burkina-Faso évolue dans un contexte économique basé désormais sur le Libéralisme économique, ce qui n'est pas toujours favorable pour nos entreprises qui n'ont pas encore atteint un certain niveau de compétitivité. Cette situation est accentuée par la conjoncture économique qui prévaut depuis plusieurs années.

Il faut ajouter à cela que notre pays qui fait partie des pays les moins avancés économiquement du monde possède une population très pauvre dont la majeure partie du revenu est destinée à la consommation.

L'assurance apparaît donc comme un bien de luxe pour la population rendant ainsi hostile et rude l'environnement économique et social de l'UAB-vie donc difficile son développement.

- **La Concurrence**

L'UAB-vie doit faire face à une très forte concurrence qui se caractérise par la présence de plusieurs sociétés d'assurances souvent plus expérimentées dans le domaine parce que plus anciennes sur le marché. Les principales sociétés d'assurance au Burkina-Faso sont :

- a. La société Nationale d'assurance et de Réassurance vie (SONAR-vie) ;
- b. La FONCIAS-vie devenue AGF Burkina Assurances vie ;
- c. La Générale des Assurances vie ;
- d. COLINA – Assurance vie ;

A ces sociétés d'assurance, il faut ajouter des sociétés de courtage qui sont ARISK Burkina, COUR RAS, GRAS SAVOYE, VINCENT & ASSOCIES ... qui partagent aussi le marché d'assurance-vie.

b. Performances

Au Burkina Faso, les compagnies d'assurances vie sont classées selon le chiffre d'affaire annuel. C'est à ce titre que nous examinons le comportement de l'UAB-vie sur le marché.

- Place de l'UAB-vie sur le marché

Comme toute entreprise, l'UAB-vie a pour objectif principal la satisfaction de la clientèle. Afin de conquérir la plus grande part de marché possible, elle a mis en place dès sa création une politique commerciale dynamique grâce à un personnel qualifié.

En une décennie d'existence, l'UAB-vie s'est faite une place de choix sur le marché des assurances vie. Sur le marché national, l'UAB-vie gère 41,05% du marché national de l'assurance vie, ce qui fait d'elle la première société d'assurance vie au Burkina-Faso.

En outre, l'UAB-vie possède le plus vaste réseau d'agents commerciaux répandus sur l'ensemble du territoire national.

Evolution du chiffre d'affaire

-Tableau du chiffre d'affaire de1998 à 2003

ANNEES	CHIFFRE D'AFFAIRES
1998	2 127 138 398
1999	1 654 470 830
2000	1 725 364 428
2001	1 730 278 490
2002	1 872 313 452
2003	2 489 557 604

Source : Service financier et comptable de l'UAB-vie 2004

I.5 Les Agences et les Produits de l'UAB-vie

a. Les Agences

L'agence est une succursale de la société qui exécute les mêmes opérations que cette dernière. Ses mouvements sont contrôlés par la société mère.

Ces différentes agences sont :

- L'agence située sur l'avenue KWAME N' KRUMA
- L'agence de la zone commerciale (Ouaga)
- L'agence de Larlé (Ouaga)
- L'agence de Koudougou
- L'agence de Bobo-Dioulasso
- L'agence de Banfora
- L'agence de Kaya
- L'agence de Pouytenga

Elles sont dirigées chacune par un agent général, personne physique exerçant une profession libérale. Il est rémunéré à la commission et est mandaté par l'Union des Assurances du Burkina-vie à laquelle il est lié par un traité de nomination.

b. Les Produits de l'UAB-vie

Dans le souci d'une entière satisfaction de la clientèle, l'UAB-vie propose une gamme variée de produits conçus en fonction des besoins spécifiques des clients ; ainsi nous pouvons citer :

- **Mixte Plus** : c'est un contrat mixte parce qu'il allie l'assurance en cas de vie (épargne) et l'assurance en cas de décès. Il garantit le paiement d'un capital quoiqu'il advienne :

- à l'assuré lui-même en cas de vie au terme du contrat ;
 - A l'assuré en cas d'invalidité totale et définitive ;
 - Au (x) bénéficiaire (s) désigné (s) dans le contrat en cas de décès de l'assuré au cours du contrat.
- **Rente Education Plus (REP)** : c'est un contrat de capitalisation, il a pour objet la scolarisation des enfants ou leur insertion dans la vie active grâce à une épargne souple et attrayante. La durée du contrat est choisie librement par l'adhérent avec un minimum de cinq (5) ans. Le contractant peut être un parent d'élèves ou toute personne qui a à sa charge des enfants.
- **Libre Epargne (LE)**: c'est un contrat de capitalisation destiné aux professions libérales afin de leur permettre de préparer leurs retraites grâce à des cotisations dont le montant et la périodicité sont à leurs choix.
- **Cauri d'or** : c'est un nouveau produit mis sur le marché dans le but d'aider le secteur informel dans son développement. Il est de deux formules :
- La première a une durée maximale de douze (12) mois et en cas de décès accidentel de l'assuré, outre l'épargne constituée, un montant de 25 000 f est versé au bénéficiaire désigné dans le contrat ;
 - Dans la seconde formule, seule la durée minimale de trois (3) ans est indiquée. Le décès accidentel de l'assuré produit un versement par la compagnie d'un capital de 100 000 F en plus de l'épargne constituée par le défunt au bénéficiaire.

Aussi bien dans l'une ou l'autre formule la cotisation minimale est de cent cinquante (150) F CFA par jour. La souscription se fait dans les kiosques installés par l'UAB-vie dans les différents marchés de Ouaga ou directement au siège de la compagnie.

- **Epargne Retraite Avenir (ERA)** : c'est un contrat de pure capitalisation dont l'objet est de se constituer une pension de retraite par épargne ou une pension complémentaire au vue de soutenir le niveau de vie à la retraite. Ce contrat garantit la retraite de l'assuré sans formalités médicales et sans limite d'âge.
- **Temporaire Décès (TD)** : c'est un contrat de prévoyance, une assurance en cas de décès. Il permet de garantir les crédits auprès des banques et autres établissements financiers, ou de prévoir un capital pour les ayants droits de

l'assuré en cas de décès de celui-ci. Sont admises à ce contrat les personnes âgées de 18 à 60 ans.

- **Epargne Logement Bayiri (ELB)**: c'est une assurance en faveur de la diaspora Burkinabé pour investir dans leur pays. Cette épargne permet à l'UAB de leur payer des parcelles ou des logements.
- **Bon Epargne scolarité tranquille (BEST)** : C'est une assurance qui permet la prise en charge de la scolarité future des enfants ou leur insertion dans la vie.

Sa cotisation minimale est de 25 000 F CFA par bon de versement autonome les uns des autres. Le contrat est destiné aux professions libérales.

II. Domaine d'étude

II.1 Le recouvrement et contentieux

Le recouvrement et contentieux est un détachement de la direction technique qui s'occupe comme son nom l'indique des questions concernant le recouvrement et le contentieux de l'UAB-vie. En effet, annuellement l'UAB-vie envoie à chaque assuré une lettre faisant état de ses cotisations. Mais avant d'envoyer il est nécessaire d'effectuer des vérifications afin de s'assurer que l'information contenue dans la lettre est juste. Cette tâche incombe au recouvrement et contentieux. Il traite également les demandes de corrections (l'assuré a constaté des erreurs sur sa lettre), de trop perçu (l'assuré a résilié son contrat et la source de paiement continu d'effectuer le prélèvement) et les demandes d'augmentation de prime. Pour les demandes de correction et de trop perçu il est nécessaire de faire certains traitements afin de prouver si les plaintes des assurés sont fondées.

II.2 Problématique

La diversité et la qualité des produits de l'UAB-vie dans le domaine de l'assurance-vie lui a valu la première place sur le plan national. Malgré ce niveau de progression le traitement du contentieux continu de souffrir d'une gestion manuelle.

En effet la procédure actuelle est exécutée manuellement rendant ainsi le travail difficile et les décisions du contentieux plus ou moins erronées.

Ces difficultés sont dues à plusieurs raisons :

- **La collecte des informations** : lorsque l'assuré constate des irrégularités sur la situation de son contrat il est nécessaire de réunir un certain nombre d'informations concernant le contrat de l'assuré. Après avoir réuni ces informations il faut alors faire une confrontation entre les irrégularités constatées par le client et les informations disposées par l'UAB-vie. A l'issue de cette confrontation l'UAB-vie pourra alors prendre position.

La difficulté majeure rencontrée est la recherche de l'information. En effet les informations sont disposées un peu partout dans la société.

- **Le nombre d'agent exécutant la procédure** : ce nombre est très restreint vu le nombre que le nombre d'assurés est élevé rendant le travail lent et fastidieux.
- **L'attribution des numéros de pièces comptable** : parmi les informations nécessaires pour trancher le contentieux il y a le référentiel contenant les numéros de pièces comptables. Ces numéros sont attribués par le service comptable et correspondent à des versements pour des mois donnés. La politique d'attribution de ces numéros n'est pas efficace. En effet plusieurs mois peuvent avoir le même numéro de pièce comptable et ceci n'optimise pas le travail.

Face ces difficultés il paraît nécessaire de mettre en place un système permettant de rendre l'intersection entre les informations fournies par le client et celles disposées par l'UAB-vie plus efficace et faciliter l'accès à l'information. Ce système doit également permettre de rendre le travail plus rapide et plus sûr afin de rendre les décisions du contentieux fiables et assurer par la même occasion la crédibilité de l'UAB-vie vis-à-vis de ses clients.

II.3 Langage et méthode d'analyse

a. Langage de modélisation : UML

En octobre 1994 le concepteur de la méthode Booch, G. Booch, et celui de la méthode OMT J. Rumbaugh décident d'unifier leurs méthodes. Ils sont rejoints une année plus tard par Ivan Jacobson le concepteur de la méthode OOSE. Ainsi est né Unified Modeling Language ou UML.

UML est un langage de modélisation fondé sur les concepts orientés objet. Elle a été conçue pour la modélisation de tous les phénomènes de l'activité de l'entreprise indépendamment des techniques d'implémentation mise en œuvre par la suite. En plus des éléments constitutifs de tout langage UML a choisit la notation graphique fondée sur des diagrammes.

➤ Les diagrammes d'UML

Pour faciliter la vision du non-programmeur et le travail du programmeur et afin d'obtenir une vue d'ensemble du système en un temps beaucoup plus court UML à initié le concept visuel fondé sur les diagrammes. UML 2.0 définit quatorze (13) diagrammes repartis selon les modèles :

- ✓ **Les modèles statiques**
 - Diagramme de classes
 - Diagramme d'objets
 - Diagramme de composants
 - Diagramme de déploiement
 - Diagramme de paquetage
 - Diagramme de structure composite

- ✓ **Les modèles dynamiques**
 - Diagramme des cas d'utilisation
 - Diagramme d'état-transitions
 - Diagramme d'activités

- ✓ **Les modèles d'interaction**
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de temps

UML n'impose pas un ordre d'utilisation des diagrammes, ces derniers sont utilisés en fonction des besoins du programmeur.

➤ **Les forces et faiblesses d'UML**

Les forces :

- UML est un langage formel et normalisé
- Gain de précision
- Gage de stabilité
- Encourage l'utilisation d'outils
- UML est un support de communication performant
- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel

Les faiblesses :

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

Même si l'expérience est une utopie, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.

Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

b. Méthode d'analyse : RUP

➤ **Présentation**

Rational Unified Process (RUP) est un processus de réalisation ou d'évolution de logiciels entièrement basé sur UML. Il est constitué d'un ensemble de directives dont l'objectif est de produire, à partir du cahier des charges, des logiciels de qualité qui satisfont les besoins des utilisateurs. Chaque directive permet de définir qui fait quoi et à quel moment, permettant ainsi de structurer les différentes étapes du projet.

Le processus RUP a l'avantage d'être piloté par les cas d'utilisation offrant ainsi un gain de temps. Mais la performance réelle du processus réside dans le fait qu'il soit itératif car un projet est divisé en une suite de sous-projets et chaque sous projet est réalisé avec les mêmes activités. Les créateurs de RUP proposent une approche pour éviter de traiter un projet entièrement avec une livraison intervenant longtemps après la rédaction du cahier des charges.

➤ **Cycle de développement**

Le cycle de développement de RUP est divisé en quatre phases :

- **la phase d'Inception** : cette phase consiste à délimiter et évaluer le projet. A l'issue de la phase d'Inception les principaux cas d'utilisation et une première ébauche de l'architecture du logiciel sont déterminés.
- **la phase d'élaboration** : cette a pour objectif d'analyser le domaine d'étude, établir une solide architecture du futur système, développer le plan et éliminer les risques du projet.
- **la phase de construction** : c'est la phase développement du logiciel conformément à l'architecture définie lors de la phase d'élaboration.
- **la phase de transition** : c'est la phase d'implantation du logiciel et la formation des utilisateurs.

Chacune des phases peut être découpée en plusieurs itérations, et comprend un ensemble d'activités.

➤ **Activités**

Chaque phase de RUP est constituée d'un ensemble d'activités. Une activité étant un ensemble d'actions décrit par un diagramme d'activité. Les principales activités de RUP sont :

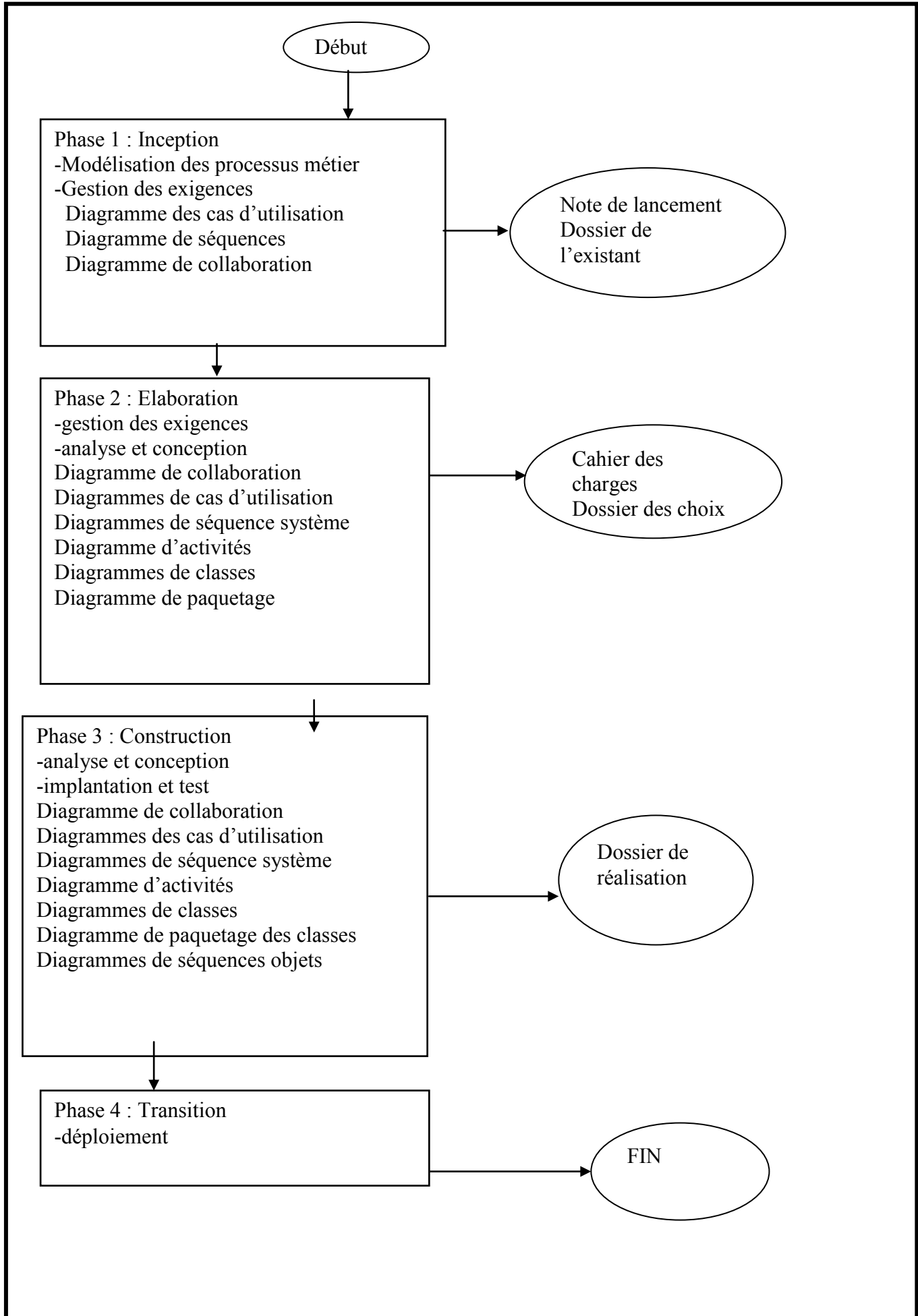
- Modélisation des processus métier,
- La gestion des exigences
- Analyse et conception
- Implantation et test
- Déploiement

Ces activités sont menées en fonction des phases du cycle de développement.

Phase	Activités
Inception	Modélisation des processus métier Gestion des exigences
élaboration	Gestion des exigences Analyse et conception
construction	Analyse et conception Implantation et test
transition	déploiement

Tableau des phases et des activités correspondantes

Le tableau suivant présente les différentes phases, les activités et les diagrammes correspondants ainsi que les livrables :



II.4 Les acteurs du projet

Les acteurs d'un projet sont répartis en fonction des rôles. On distingue trois groupes repartis comme suit :

- ✓ **le groupe de pilotage** : chargé de prendre les grandes décisions concernant l'orientation, il définit les délais à respecter et les moyens à mettre en œuvre pour atteindre les objectifs. Les membres du groupe de pilotage sont :
 - M SORGHO Daouda (chef du service informatique de l'UAB-vie),
 - M OUEDRAOGO Raphaël (responsable du recouvrement et contentieux),
 - M BANDE Moustapha (directeur des politiques de promotion de TIC).
- ✓ **le groupe de projet** : il est chargé de l'étude, la conception et la réalisation du projet. Il est constitué de :
 - SINARE Judith et TAPSOBA Abdoul-malick tous deux étudiants en fin de cycle (CITI) à l'ESI.
- ✓ **le groupe des utilisateurs** : ce sont les utilisateurs du produit fini. Ils sont chargés de fournir les informations nécessaires à la conduite du projet. Il est composé de :
 - service de recouvrement et du contentieux
 - service informatique

II.5 Planning du stage

phase	Dossier à fournir	période	durée
Inception	Note de lancement Dossier de l'existant	9 Août au 6 septembre	4 semaines
élaboration	Cahier des charges Dossier des choix	7 septembre au 12 octobre	5 semaines
construction	Dossier de réalisation	13 octobre au 3 novembre	3 semaines
transition	Guide de l'utilisateur	4 novembre au 6 novembre	1 semaine

B. ETUDE DE L'EXISTANT

I. Objectif

L'étude de l'existant est réalisée à travers la phase d'Inception de la méthode RUP. C'est la première phase de la démarche préconisée par RUP. Elle prend acte de la naissance du projet et définit sa portée. L'objectif réel de cette phase est de prendre la décision de faire ou de ne pas faire le projet sur la base d'orientations générales claires.

Au cours de cette phase les principaux cas d'utilisations sont définis ainsi qu'une ébauche de l'architecture du futur système.

Au cours de cette phase deux activités principales sont menées. Il s'agit de la modélisation du processus métier et de la gestion des exigences. La modélisation du processus métier a pour principal objectif de s'assurer que les clients et les développeurs ont une compréhension commune du problème à résoudre. La gestion des exigences quand à elle permet aux développeurs de prendre en compte les désirs des utilisateurs pour le futur système.

II. Démarche d'analyse

Cette phase d'étude d'opportunité ou phase d'Inception va se dérouler à travers deux activités essentielles :

- **première activité : la modélisation du processus métier**

C'est la première activité menée dans la phase d'Inception. Cette activité a pour but de permettre aux développeurs et aux clients d'avoir le même niveau de compréhension du système à informatiser et permettre de dériver les besoins du futur système.

- **deuxième activité : la gestion des exigences**

Cette activité permet de décrire de façon claire les besoins fonctionnels et techniques du futur système. Cette phase ne permet pas de prendre en compte toutes les exigences du futur système car de nombreux besoins peuvent survenir au cours du temps. Cette phase est alors réitérée dans toutes les phases de RUP.

III. Modélisation du processus métier

Comprendre la structure à étudier est une étape essentielle dans le processus d'informatisation. La modélisation métier a pour but de permettre cette compréhension. Dans notre cas elle a pour objectifs de :

Comprendre le fonctionnement du contentieux

De s'assurer que le groupe de pilotage, les utilisateurs et le groupe de projet ont une compréhension commune du contentieux

Permettre de dériver l'expression des besoins du contentieux

Nous utiliserons les diagrammes de cas d'utilisations et de séquences pour illustrer cela.

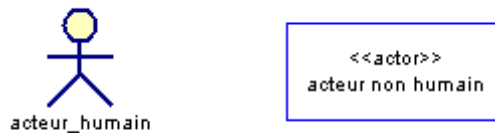
III.1 Diagramme des cas d'utilisations

Très souvent le groupe d'utilisateurs et le groupe de pilotage ne sont pas des informaticiens. Il faut donc un moyen simple et efficace pour exprimer leurs besoins. C'est le rôle joué par le diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation permet de recueillir, d'analyser et d'organiser les besoins des utilisateurs et de recenser les grandes fonctionnalités du système. Il capture le comportement d'un système tel qu'un utilisateur le voit de l'extérieur. Il scinde les fonctionnalités du système en unités cohérentes appelées cas d'utilisation ayant un sens pour les acteurs.

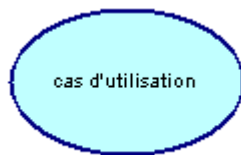
➤ Concepts et formalismes du diagramme de cas d'utilisation

- **acteur** : un acteur est une abstraction d'un rôle joué par une personne externe, un processus ou une chose et qui interagit avec le système. Il est représenté par un petit bonhomme dans le cas où l'acteur est un humain et par un classeur dans le cas contraire.



- **cas d'utilisation :**

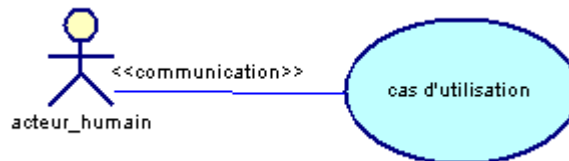
Un cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur. Il modélise un service rendu par un acteur au système. Il se représente par une ellipse contenant le nom du cas d'utilisation



Représentation graphique d'un cas d'utilisation

- **relation dans les diagrammes de cas d'utilisation :**

Relation d'association : c'est un chemin de communication entre un acteur et un cas d'utilisation et est représenté par un trait continu.

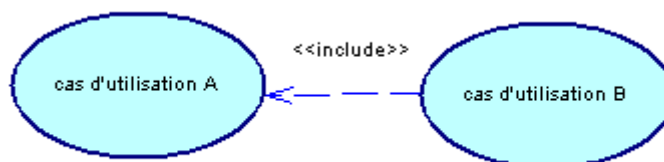


Représentation d'une relation entre acteur et cas d'utilisation

- ✓ **relation d'inclusion :**

Un cas d'utilisation A inclut un cas d'utilisation B si pour exécuter A, B est obligatoirement exécuté. Cette dépendance est représentée par le stéréotype <<include>>.

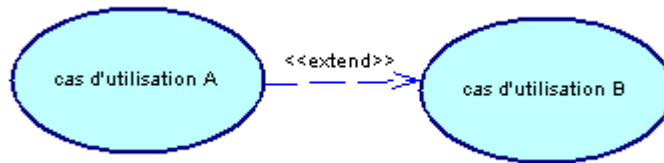
Les relations d'inclusion sont utilisées pour factoriser une partie de la description d'un cas d'utilisation qui serait commune à d'autres cas. Elles permettent aussi de décomposer un cas d'utilisation complexe en sous-cas plus simple.



Représentation d'une relation d'inclusion

✓ **relation d'extension :**

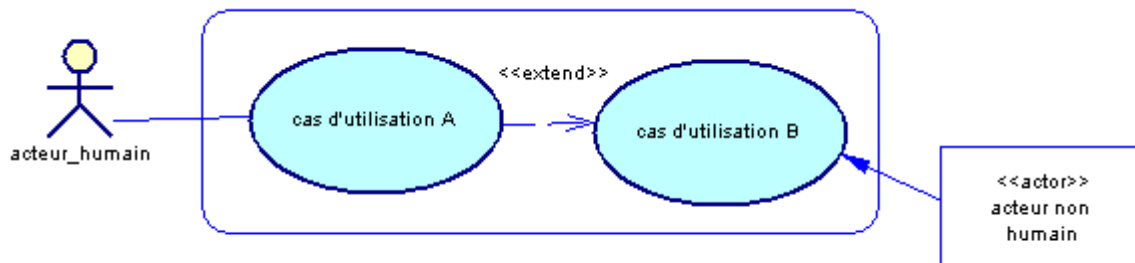
Un cas d'utilisation A étend un cas d'utilisation B si le cas d'utilisation A peut être appelé au cours de l'exécution de B. cette dépendance est représenté par le stéréotype <<extend>>.



Représentation d'une relation d'extension

• **représentation du diagramme de cas d'utilisation :**

Le système est représenté par un cadre dans lequel figure le nom du système. Les acteurs sont à l'extérieur du cadre et les cas d'utilisation à l'intérieur.

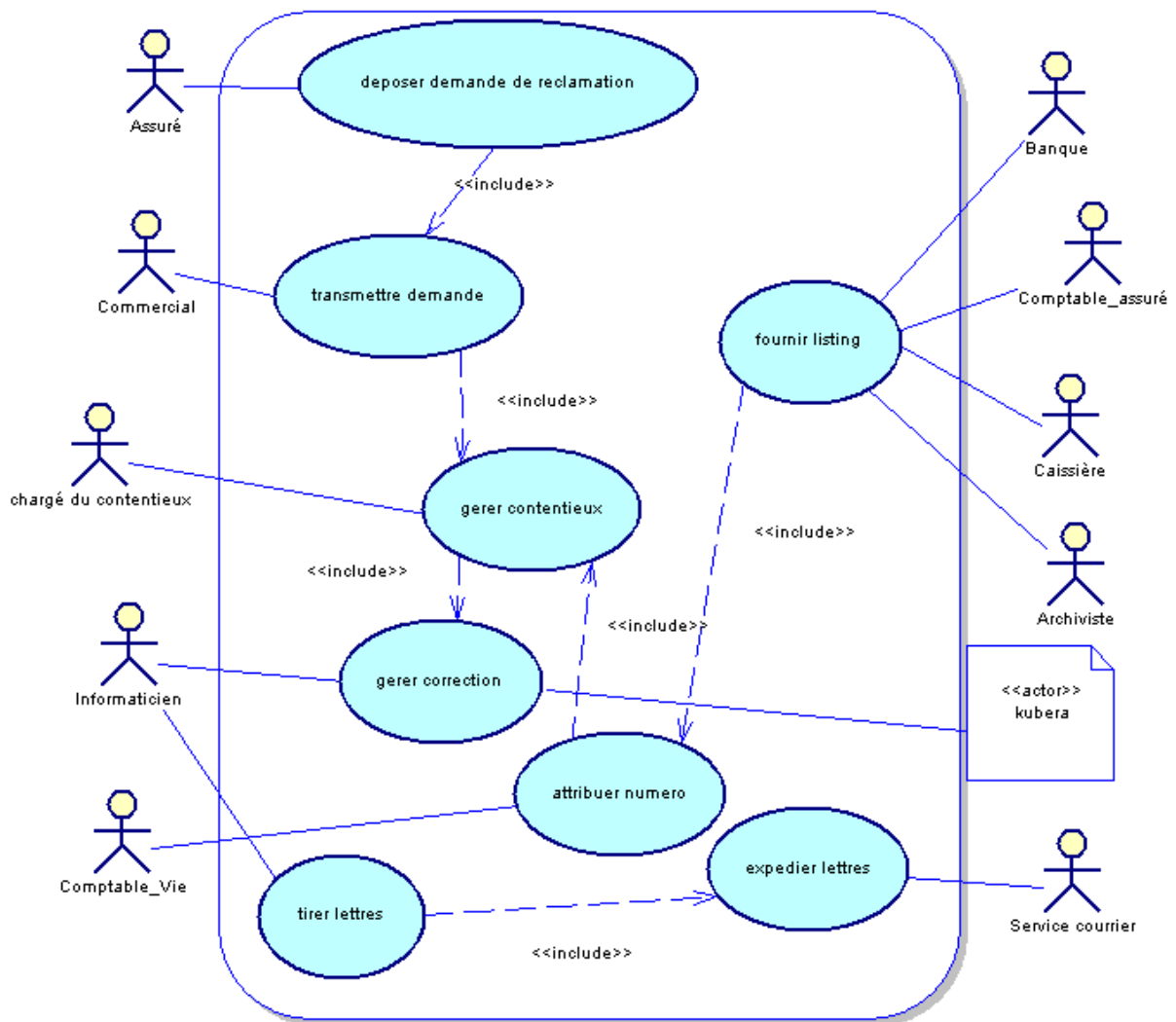


Représentation d'un diagramme de cas d'utilisation

➤ **Les acteurs et les cas d'utilisation**

- **assuré** : déposer demande
- **service clientèle et après vente** : transmettre demande
- **chargé du contentieux** : gérer contentieux
- **informaticien** : gérer correction, tirer lettres
- **comptable-vie** : fournir référentiel
- **comptable-assuré** : fournir listing
- **banque** : fournir listing
- **caissière** : fournir listing
- **archiviste** : fournir listing
- **service courrier** : expédier lettres
- **kubera** : gérer correction

➤ **Diagramme de cas d'utilisation correspondant :**



➤ **Description textuelle des cas d'utilisation**

Les diagrammes de cas d'utilisation décrivent les grandes fonctionnalités du système du point de vue des acteurs, mais n'expose pas de façon détaillée la communication entre acteurs et cas d'utilisation. Il est alors recommandé de faire une description textuelle des différents cas d'utilisation.

La description textuelle se compose de trois parties :

La première partie permet d'identifier le cas d'utilisation et contient les informations suivantes :

Nom : nom du cas d'utilisation, en générale on utilise des verbes à l'infinitif

Résumé : définit d'une façon succincte l'objectif du cas d'utilisation

Acteurs : contient ceux qui réalisent le cas d'utilisation et ceux qui utilisent les résultats du cas d'utilisation.

Date : date de création et de mise à jour de la description

Responsable : responsable de la description textuelle

Version : version de la description courante

La deuxième partie contient la description du fonctionnement des cas d'utilisation sous forme d'une séquence de messages échangés entre les acteurs et le système. La deuxième partie contient les informations suivantes :

Préconditions : décrivent l'état dans lequel doit être le système avant le déclenchement du cas d'utilisation.

Scénarii : les scénarii sont décrits sous la forme d'échanges d'évènements entre le système et les acteurs du cas d'utilisation. On distingue le scénario nominal qui décrit un comportement normal du système et le scénario d'exception qui décrit un cas d'erreur. Le scénario alternatif est une variante du scénario nominal.

Postconditions : décrivent l'état du système à l'issue des différents scénarii.

La troisième partie est optionnelle et contient des spécifications non fonctionnelles.

➤ **Descriptions des cas d'utilisations du contentieux**

CU1 : Cas d'utilisation : déposer demande de réclamation

Résumé : permet à l'assuré d'entamer une procédure de réclamation lorsqu'il constate des irrégularités sur la situation de son contrat.

Acteurs : assuré, SCV, chargé du contentieux

Préconditions : avoir établi au moins un contrat avec l'UAB-vie et avoir constater des erreurs sur sa lettre.

Scénario nominal :

1. L'assuré rédige la demande
2. l'assuré transmet sa demande au commercial

Scénario alternatif :

1. demande incorrecte : il faut répartir au point 1 du scénario nominal

CU2 : Cas d'utilisation : transmettre demande

Résumé : lorsque l'assuré dépose sa demande de réclamation ce cas d'utilisation permet au commercial de transmettre la requête au chargé du contentieux.

Acteurs : SCV, chargé du contentieux

Préconditions : le commercial a reçu la demande de l'assuré

Scénario nominal :

1. l'agent transmet la demande

CU3 : Cas d'utilisation : gérer contentieux

Résumé : ce cas d'utilisation permet au chargé du contentieux après réception de la demande de réclamation de faire une confrontation entre les erreurs constatées par l'assuré et les informations disposées par l'UAB-vie sur le contrat de l'assuré. Il permet de départager le client et l'UAB-vie en partant sur l'hypothèse que l'information fournie par le client est vraie.

Acteurs : chargé du contentieux, informaticien

Préconditions :

1. recevoir la demande de réclamation de l'assuré
2. disposer du référentiel de numéros de pièces comptables
3. disposer de la situation du contrat de l'assuré
4. disposer de l'état des encaissements de l'assuré
5. disposer du récapitulatif des listes envoyées par l'organisme payeur.

Scénario nominal :

Pointage : cette opération consiste à faire une intersection entre les numéros de pièces comptables figurants sur le référentiel et ceux figurants sur l'état des encaissements de l'assuré. Le pointage permet de recenser les numéros de pièces comptables qui figurent sur le référentiel et qui ne figure pas sur la situation des encaissements de l'assuré. Ces numéros feront l'objet d'une vérification dans l'étape suivante.

Vérification : connaissant les numéros manquants, le nom de l'assuré, la société dans laquelle travaille l'assuré et les mois concernés, le chargé du contentieux procède à une vérification. Pour chaque mois il vérifie si le nom de l'assuré figure dans le listing du mois. A l'issue de cette vérification on aura les alternatives suivantes :

6. les mois payés qui n'ont pas été enregistré
7. les mois impayés
8. la source des autres irrégularités

La confrontation : cette opération consiste à faire une confrontation entre les résultats de la vérification et les informations contenues dans la demande de réclamation de l'assuré. Après cette confrontation il pourra enfin dire si l'assuré a raison ou pas. Au cas où il a raison un rapport contenant les corrections à porter sur le contrat de l'assuré est remis à l'informaticien pour faire la mise à jour.

Scénario alternatif :

Manque d'informations nécessaires : dans ce cas le chargé du contentieux attend de réunir toutes les informations nécessaires et reprendre le processus depuis le pointage.

CU4 : Cas d'utilisation : gérer correction

Résumé : ce cas d'utilisation permet à l'informaticien de corriger les erreurs détectées pendant la gestion du contentieux.

Acteurs : informaticien, Kubera

Préconditions : disposer du rapport du contentieux

Scénario nominal :

1. l'informaticien demande à se connecter
2. Kubera demande le login et le mot de passe
3. l'informaticien fournit le login et le mot de passe
4. Kubera lui présente les fonctionnalités nécessaires
5. l'informaticien fait la mise à jour

6. L'informaticien se déconnecte

Scénario alternatif :

Login ou mot de passe incorrecte : dans ce cas il faut repartir au point 3 du scénario nominal.

CU5 : Cas d'utilisation : fournir listing

Résumé : ce cas d'utilisation permet à la banque ou au Comptable-assuré d'envoyer la liste des assurés ayant autorisés le prélèvement ou le virement. Ce cas d'utilisation est utilisé également au sein de l'UAB-vie notamment par la caissière pour envoyer la liste des assurés qui ont payés par espèce ou par chèque et par l'archiviste pour fournir les listings qui ont été archivés. En effet lorsque le recouvrement et contentieux finit de traiter un listing ce listing est archivé.

Acteurs : banque, Comptable-vie, Comptable-assuré, caissière, archiviste.

Préconditions : les assurés ont autorisés le prélèvement ou ont payer par espèce à la caisse.

Scénario nominal :

1. l'organisme payeur reçoit l'ordre de payer la prime mensuelle
2. l'organisme établit la liste des assurés
3. il transmet la liste à l'UAB-vie

Scénario alternatif :

Le compte de l'assuré n'est pas approvisionné : dans ce cas il faut avertir l'assuré.

CU6 : Cas d'utilisation : fournir référentiel

Résumé : ce cas d'utilisation permet au comptable d'attribuer des numéros de pièces comptables aux différents versements et de transmettre les listings au service informatique pour l'encaissement.

Préconditions : le Comptable-vie a reçu les listings

Scénario nominal :

1. le comptable réceptionne le listing et le chèque
2. il vérifie le listing
3. il attribut les numéros de pièces comptables
4. il transmet les listings au service informatique pour l'encaissement

Scénario alternatif :

Le listing est incorrect : dans ce cas il faut répartir au point 1 du scénario nominal

CU7 :Cas d'utilisation : tirer lettres

Résumé : ce cas d'utilisation permet à l'informaticien de tirer la lettre de l'assuré après la correction.

Acteurs : informaticien, Kubera, service courrier

Préconditions : des corrections ont été faites sur le contrat de l'assuré

Scénario nominal :

1. l'informaticien demande à se connecter à Kubera
2. Kubera demande le login et le mot de passe
3. l'informaticien fournit le login et le mot de passe
4. Kubera lui présente les fonctionnalités nécessaires
5. l'informaticien lance l'impression
6. l'informaticien se déconnecte

CU8: Cas d'utilisation : expédier lettre

Résumé : après le tirage des lettres par le service informatique les lettres sont envoyées au service courrier qui se charge de transmettre les lettres aux assurés.

Acteurs : service courrier, assuré

Préconditions : le service courrier a reçu les lettres

Scénario nominal :

1. le service courrier vérifie les lettres
2. le service courrier enveloppe les lettres
3. le service courrier expédie les lettres aux assurés

Scénario alternatif :

La lettre est incorrecte : dans ce cas il faut repartir au point 1 du cas d'utilisation précédent.

L'adresse est incorrecte : tirer la lettre avec la bonne adresse

III.2 Diagramme de séquence

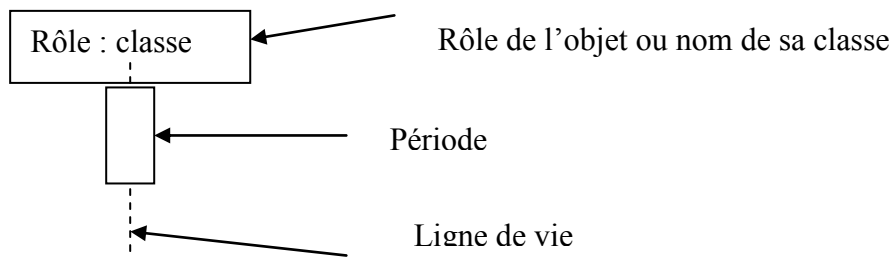
Un diagramme de séquence permet de modéliser la dynamique d'un système.

Il décrit les interactions entre un groupe d'objets en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets. Il peut également montrer les flux de données échangés lors des envois de message.

Dans notre cas le diagramme de séquence vont nous permettre de décrire la dynamique de certains cas d'utilisation.

➤ Concepts et formalismes

- **Lignes de vie** : une ligne de vie montre les actions et les réactions, ainsi que les périodes pendant lesquelles un objet du système est actif, c'est-à-dire qu'il exécute une fonction donnée.

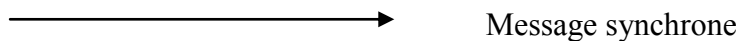


- **Envoi de message :**

L'envoi d'un message est représenté par une flèche horizontale reliant la ligne de vie de l'émetteur à la ligne de vie du destinataire. Cependant il existe plusieurs types de message. On a :

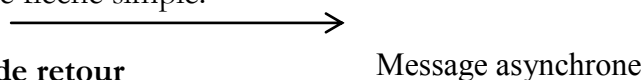
1. les messages synchrones

Il est le plus utilisé. Un message est dit synchrone si l'expéditeur attend que la méthode invoquée chez le destinataire soit terminée avant de continuer son activité. Le message synchrone bloque l'expéditeur jusqu'à l'obtention d'une réponse venant du destinataire. Il est représenté par une flèche pleine.



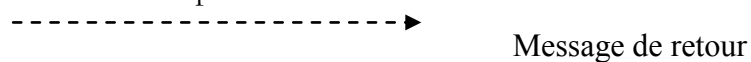
2. les messages asynchrones

Contrairement au message synchrone, le message asynchrone ne bloque pas les activités de l'expéditeur. Ce dernier peut mener d'autres activités en parallèle. Il est représenté par une flèche simple.



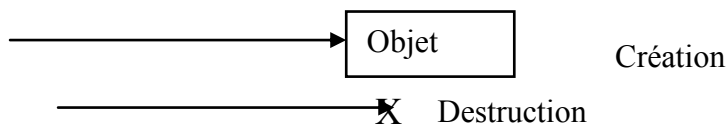
3. les messages de retour

Un message retour est une réponse à un message synchrone ou asynchrone. Il est représenté par une flèche en pointillé



4. les messages de création et de destruction

Ces types de messages donnent naissance à une nouvelle ligne de vie (création) ou mettent fin à une ligne de vie d'un objet (destruction).



5. les messages réflexifs

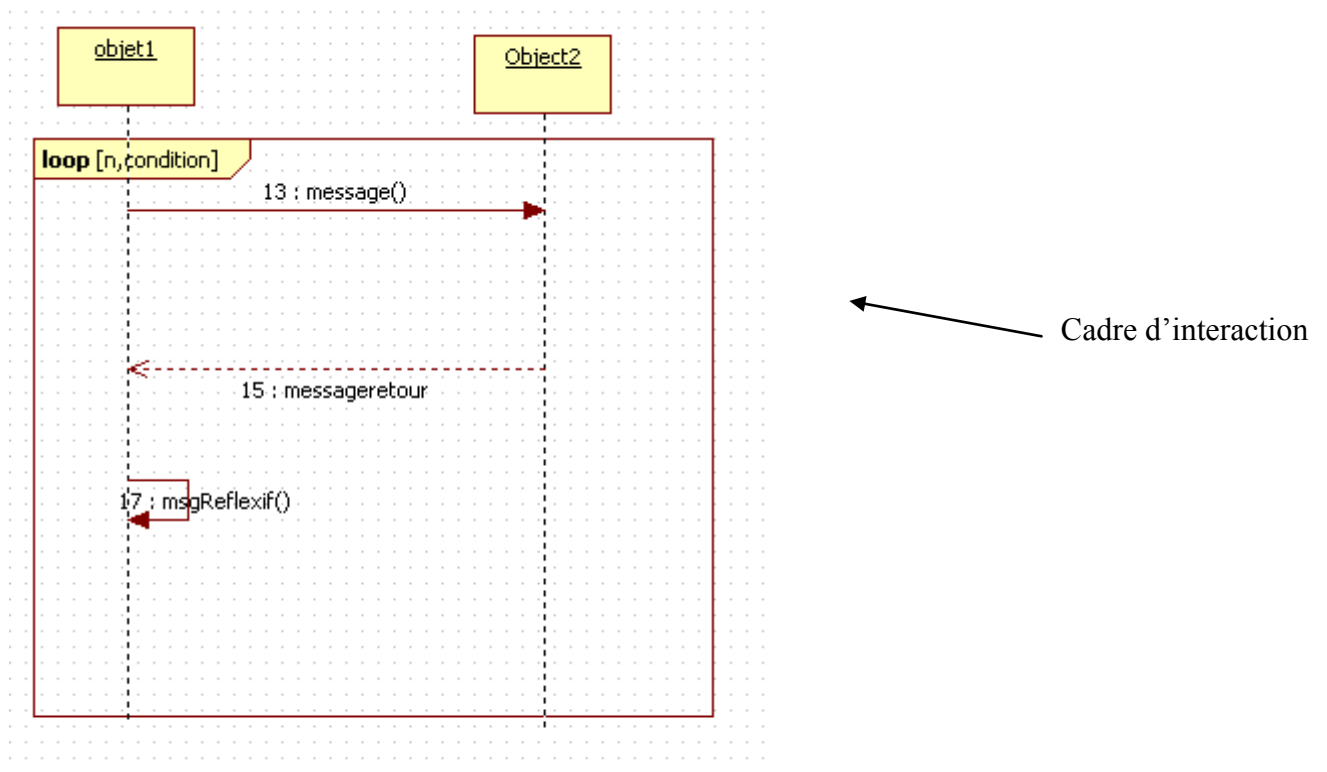
Un objet peut s'envoyer un message. Un tel message est dit réflexif



- **Les cadres d'interaction :**

Un cadre d'interaction est une partie du diagramme de séquence associé à une étiquette. Elle contient un opérateur qui détermine les modalités d'exécution. Les principales modalités sont :

- **L'alternative** : elle s'obtient en utilisant l'opérateur **opt** suivi d'une condition de test. Si la condition vérifiée, le contenu du cadre est exécuté.
- **La boucle** : la boucle est réalisée par l'opérateur **loop** suivi des paramètres min et max et d'une condition de test. Cela implique que le cadre est exécuté min tant que la condition est vérifiée et que le nombre d'exécution maximale ne vaut pas max.



Représentation d'un diagramme de séquence

➤ Représentation de quelques diagrammes de séquences

Diagramme de séquence du cas d'utilisation gérer contentieux

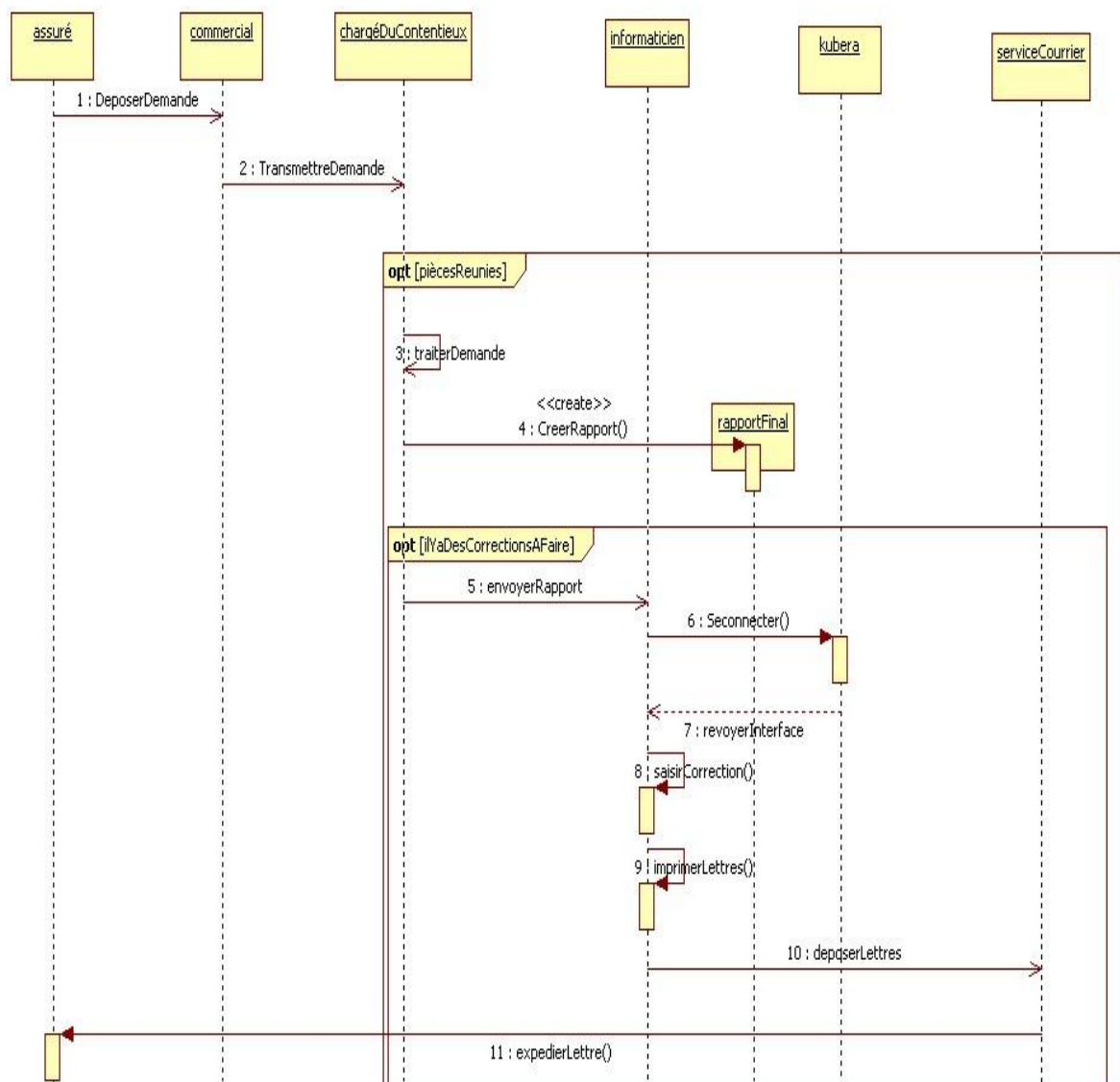


Diagramme de séquence du traitement interne du contentieux

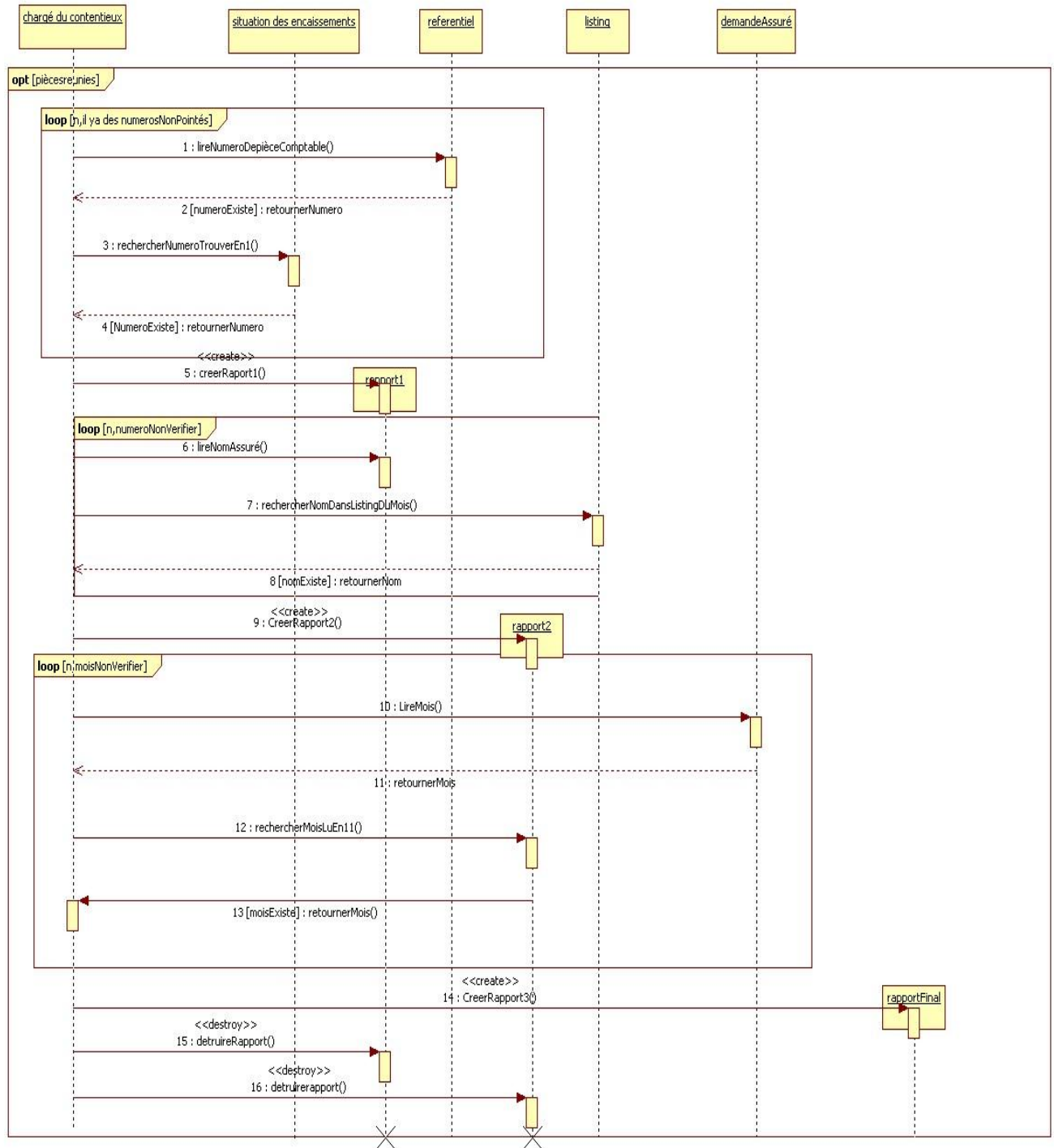


Diagramme de séquence du cas d'utilisation gérer correction

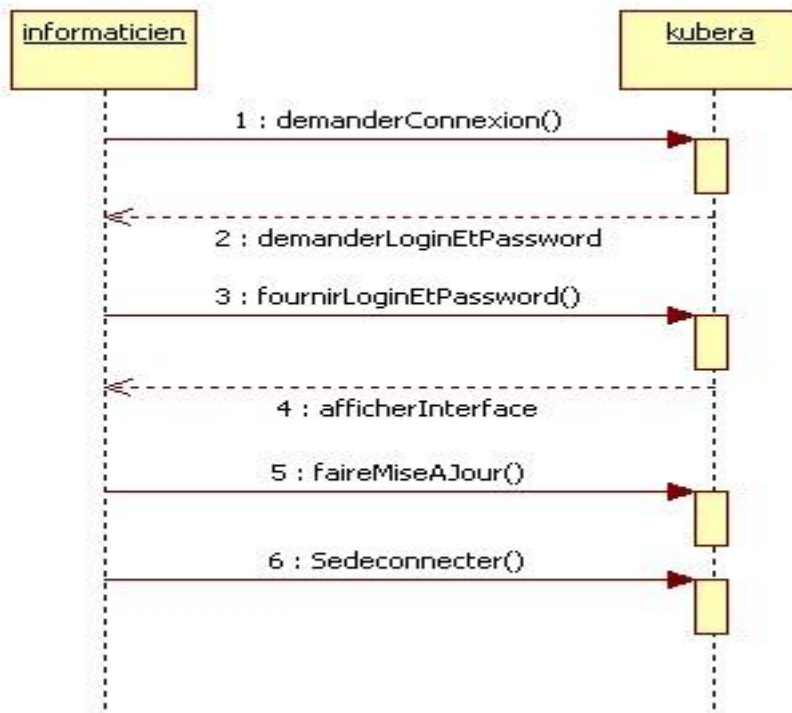
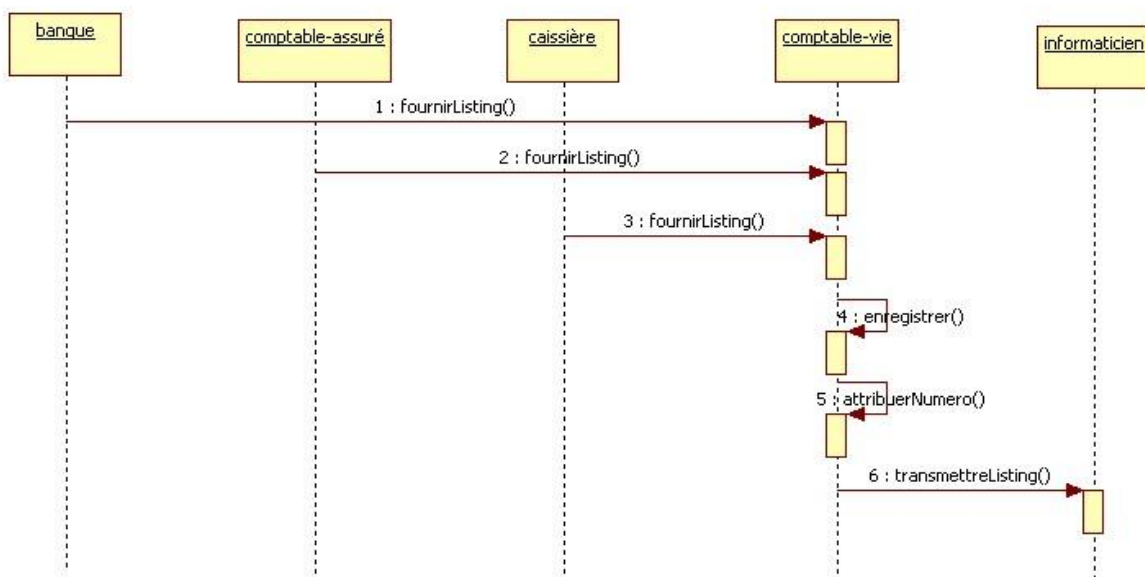


Diagramme de séquence du cas d'utilisation attribuer numéros



IV. Gestion des exigences

Connaître toute les exigences des utilisateurs sur le futur système est très important dans un processus d'informatisation. La gestion des exigences poursuit les objectifs suivants :

- arriver à un accord avec le client et les utilisateurs sur ce que le système doit être amené à faire ;
- Fournir aux développeurs une meilleure compréhension des exigences sur le système ;
- Délimiter le périmètre du système ;
- Fournir une base pour la planification du contenu technique et fonctionnel des itérations ;
- Définir l'interface homme-machine du système.

Pour exprimer ces exigences nous utilisons le diagramme de communication appelé diagramme de collaboration dans UML 1.

IV.1 Besoins fonctionnels

a. Repérage du domaine d'étude

➤ Délimitation du projet

Connaître les différents acteurs d'un système est une étape capitale dans le processus de développement d'un logiciel. Concernant notre projet, à l'issue des interviews réalisées nous avons pu déterminer les acteurs et les différentes interactions entre ces acteurs. Ces interactions sont représentées dans le diagramme de communication que nous proposons après avoir cité les acteurs et leurs rôles.

➤ Les principaux acteurs

En UML est considéré comme acteur tout ce qui est en dehors du système à modéliser et qui interagit avec lui. Dans le cas du contentieux nous avons les acteurs suivants :

- **P'assuré** : l'assuré est l'élément déclencheur du processus du contentieux. Après avoir détecté les erreurs sur la situation de son contrat celui-ci engage un processus de réclamation. Pour cela il rédige une demande sur laquelle figure les erreurs et transmet ensuite la demande au service commercial.
- **le SVC**: le SVC est l'intermédiaire entre l'assuré et le reste des acteurs. Celui-ci est chargé de transmettre la requête de l'assuré au recouvrement et contentieux.
- **le chargé du contentieux** : il est chargé de traiter la requête de l'assuré afin de pouvoir établir la vérité. Pour cela il réunit un certain nombre de documents concernant le contrat de l'assuré pour faire la vérification.
- **le Comptable-vie** : chaque mois la banque ou la société envoie la liste des assurés ayant cotisés pour le mois. Le Comptable-vie est chargé d'enregistrer les versements et de les attribuer des numéros de pièce comptable. La liste et les numéros de pièce comptable sont utilisés dans le processus du contentieux.

- **informaticien** : après le travail effectué par le chargé du contentieux, s'il y a contentieux c'est-à-dire s'il y a des corrections à faire l'informaticien est chargé de porter les corrections. Egalement lorsque les banques ou les sociétés envoient la liste des assurés ayant cotisés l'informaticien est sollicité pour faire les encaissements au niveau de la base de données.
- **banque** : chargé d'envoyer la liste des assurés ayant autorisés le virement bancaire ainsi que le montant du virement.
- **comptable-assuré** : dans le cas ou le mode de paiement est le prélèvement à la source, le comptabilité-assuré est chargé de d'envoyer le chèque et la liste des assurés ayant autorisés le prélèvement.
- **caissière** : elle est chargée d'enregistrer les paiements en espèce ou par chèque et d'envoyer la liste au Comptable-vie.
- **archiviste** : lorsque le chargé du contentieux ne retrouve pas certains documents (en général les listings) il fait appel à l'archiviste.
- **service courrier** : lorsque le service informatique tire les lettres des assurés le service courrier est chargé de faire parvenir les lettres aux assurés.
- **kubera** : c'est un logiciel utilisé par l'informaticien pour faire les encaissements et les corrections.

b. Diagramme de communication

Un diagramme de communication (diagramme de collaboration dans UML 1) rend compte de l'organisation spatiale des acteurs à l'interaction. Il est souvent utilisé pour illustrer un cas d'utilisation ou pour décrire une opération.

➤ Concepts et formalismes

- **les lignes de vie** :

Les lignes de vie représentent les acteurs. Elles sont représentées par des rectangles contenant une étiquette dont la syntaxe est :

[<Nom_du_role>] : [<nom_du_type>].

Au moins un des deux noms doit être spécifié dans l'étiquette. Les deux points (:) Quand à eux obligatoire.

[<Nom_du_role>] : [<nom_du_type>]

Représentation graphique d'une ligne de vie

- **les connecteurs :**

Les connecteurs sont des relations entre les lignes de vie. Ils se représentent par un trait plein reliant deux lignes de vie.

- **message :**

Un message représente une information échangée entre deux lignes de vie. Le sens du message est spécifié par une flèche pointant vers l'un ou l'autre des objets de l'interaction.

La syntaxe d'un message est la suivante :

[['Cond']] [seq] [*[] ||] ['iter'] :] [R :=] msg ([par])

Cond : est une condition sous forme d'expression booléenne entre crochets.

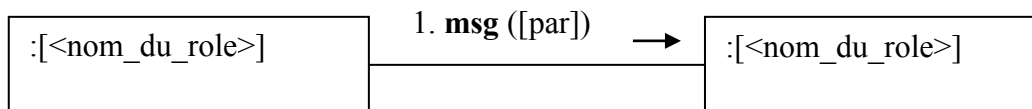
Seq : est le numéro séquentiel du message

Iter : spécifie l'envoi séquentiel

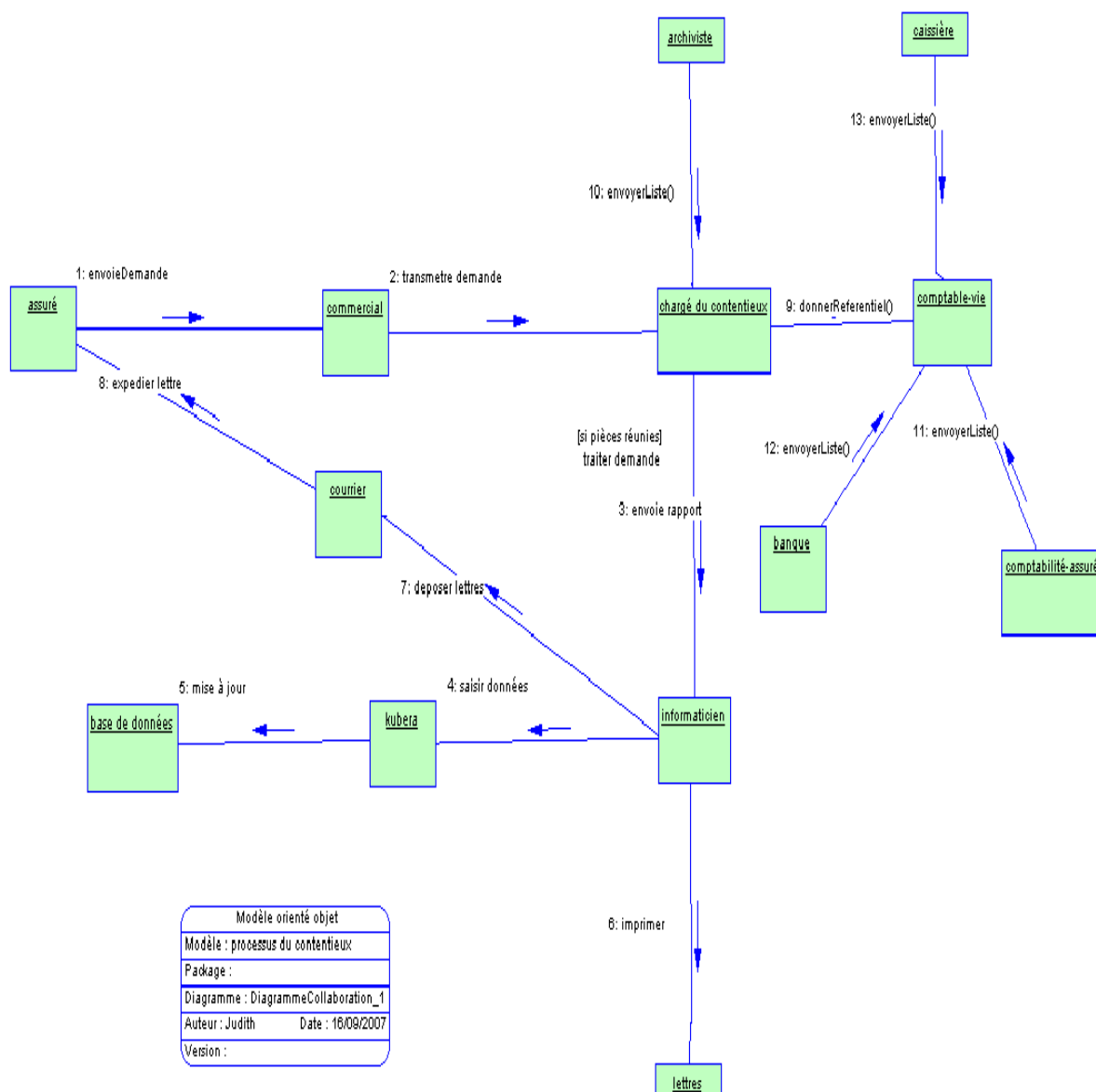
R : est une valeur de retour du message.

Msg : est le nom du message

Par : désigne les paramètres



➤ **Représentation du diagramme de communication**



c. Recueil des informations

➤ Interviews

Conformément aux objectifs poursuivis par la gestion des exigences il est nécessaire de recueillir les avis du groupe d'utilisateurs et de pilotage sur les fonctionnalités du futur système.

Dans notre cas précis, pour avoir une idée des besoins fonctionnels et dégager les différentes procédures du contentieux nous avons travaillé pendant deux (02) semaines au recouvrement et contentieux. Ainsi nous avons avec l'aide des futurs utilisateurs dégagé les besoins suivants :

Le futur système doit résoudre le problème de la recherche d'information. En effet la recherche d'information est très difficile et occupe plus de 3 / 4 du temps mis pour la résolution du contentieux.

Il doit résoudre le contentieux en partant du fait que les informations fournies par l'assuré sont vraies jusqu'à preuve de contraire.

Il doit pouvoir assurer un contrôle rigoureux des encaissements. En effet la plus part des erreurs surviennent lors de la saisie. Nous avons relevé plusieurs cas dont les plus fréquents sont :

- l'informaticien a sauté lors de la saisie le nom de l'assuré
- la prime mensuelle n'a pas été répartie au cas ou l'assuré a plusieurs contrats
- erreur pendant l'annulation

Il doit permettre au commercial de transmettre la requête de l'assuré.

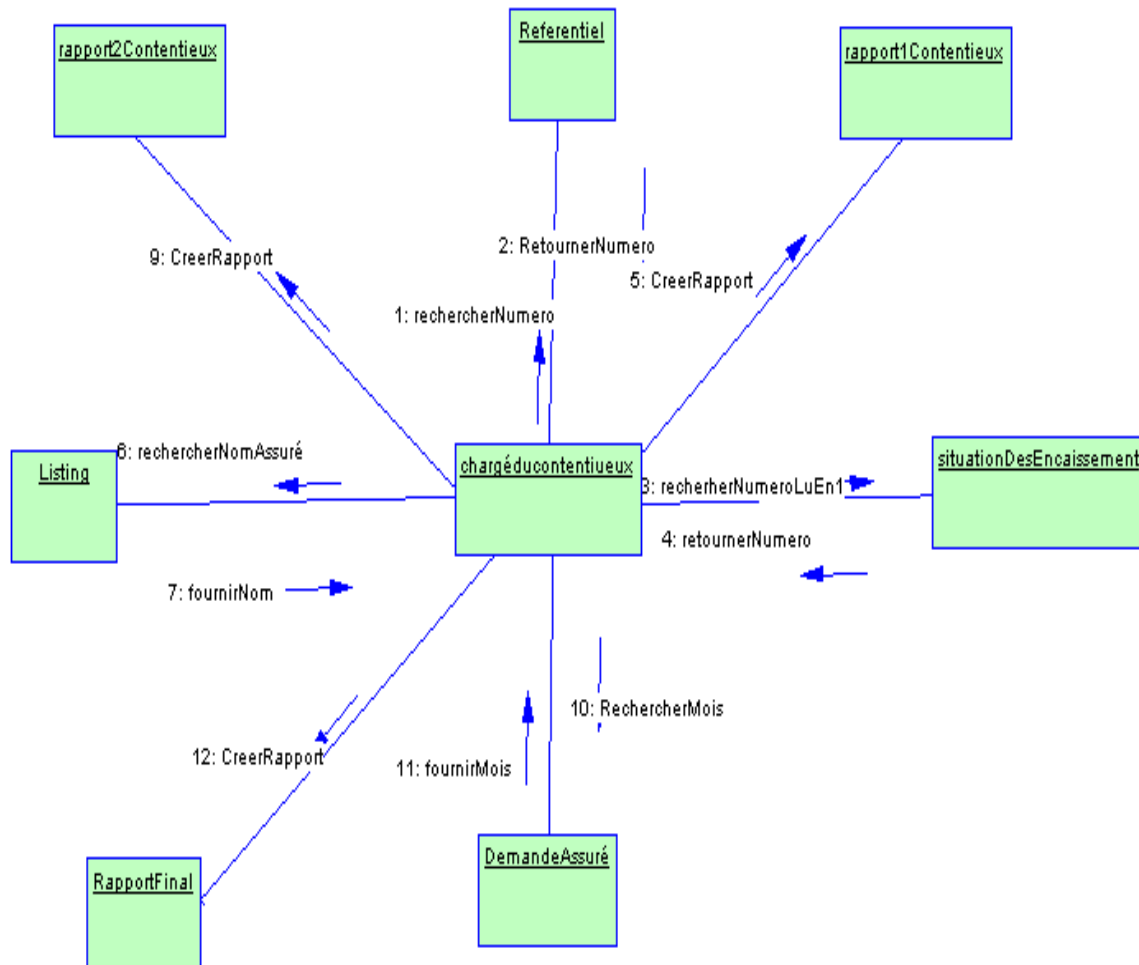
Améliorer la politique d'attribution des numéros de pièce comptable. En effet la politique actuelle rend le travail difficile car nous avons des cas ou plusieurs mois ont le même numéro de pièce comptable.

➤ Documents utilisés

Les différents documents utilisés sont :

- **listing** : c'est un document contenant l'ensemble des assurés qui ont cotisés pour un mois donné. Dans un cas de prélèvement à la source la société fait une liste de tous ses employés ayant autorisés le prélèvement et envoie la liste à la comptabilité-vie. Il en de même pour tous les assurés ayant autorisés le virement bancaire et appartenant à la même banque. Dans un cas de paiement en espèce ou par chèque par l'assuré lui-même la liste est envoyée par la caissière.
- **le référentiel** : document contenant les mois de l'année et les numéros de pièce comptable correspondant par société. Le référentiel est fourni par le Comptable-vie.
- **situation des encaissements de l'assuré** : c'est un document qui contient le récapitulatif de tous les versements effectués par le client, les numéros de pièces comptables correspondant à chaque versement, le montant et le mode de paiement. Il mentionne également le montant total des versements depuis la date d'effet du contrat jusqu'à la date de calcul.
- **lettre** : contient les mêmes informations que la situation des encaissements à la différence que la lettre ne mentionne pas les numéros de pièces comptables et le mode de paiement.
- **la demande de l'assuré** : c'est le document fourni par l'assuré. Il contient les différentes erreurs détectées par l'assuré.

➤ **Diagramme de communication correspondant**



IV.2 Besoins techniques

a. Architecture logicielle

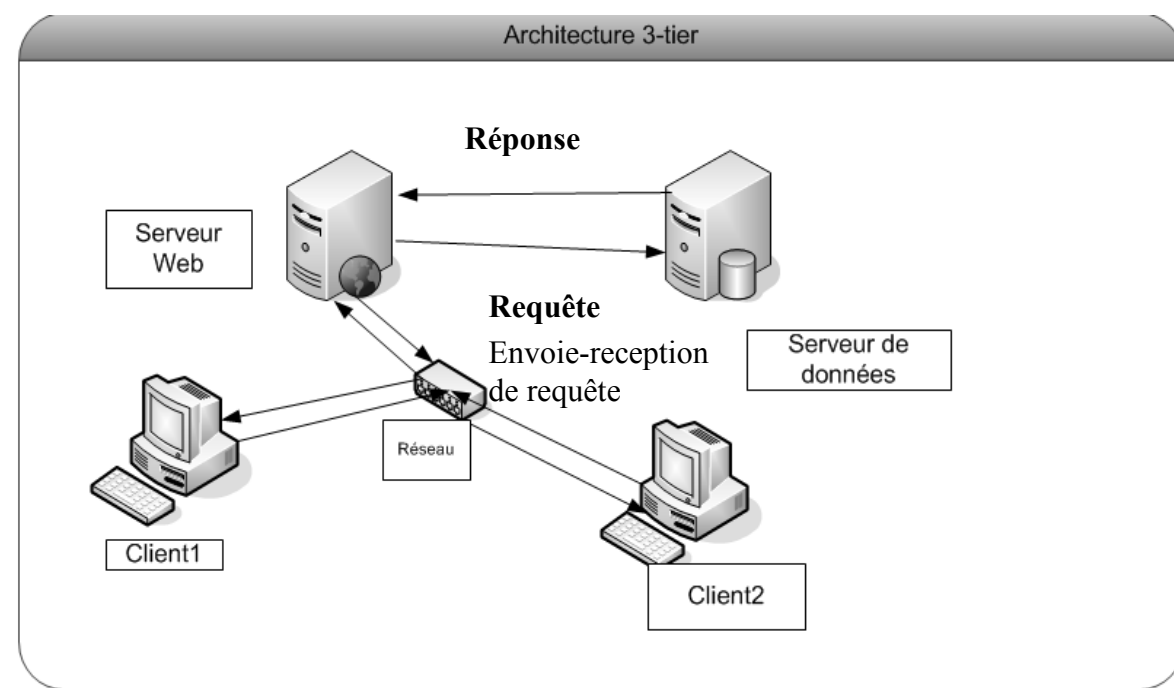
Un système est un ensemble d'objets qui coopèrent. Il possède une structure et constitue un tout organique. L'architecture décrit la structure d'un système et élaborer l'architecture permet de la concevoir. Elle décrit la structure globale d'une application.

L'architecture préconisée par l'UAB-vie pour l'automatisation du contentieux est l'architecture a trois niveaux où architecture 3-tier.

➤ Définition de l'architecture 3-tier

L'architecture 3-tier ou architecture à trois niveaux est une architecture de type client /serveur. C'est un système dans lequel l'utilisateur de l'application appelé client demande une ressource à un serveur d'application appelé middleware. Le serveur d'application fournit la ressource mais en faisant appel à un serveur secondaire généralement un serveur de base de données. Donc l'architecture 3-tier distingue 3 niveaux distincts au sein d'une application ou d'un système. Chaque niveau ayant un rôle bien précis.

Représentation de l'architecture 3-tier



➤ Avantages de l'architecture 3-tier

L'architecture 3-tier présente plusieurs avantages pouvant se résumer en trois (03) points :

- la souplesse et la flexibilité
- une sécurité accrue car elle est définie par niveau.
- une grande performance car les tâches sont partagées.

➤ Inconvénients de l'architecture 3-tier

- une expertise de développement à acquérir qui semble plus longue que dans le cadre d'une architecture 2-tiers.
- les coûts de développements d'une architecture 3-tiers sont plus élevés que pour du 2-tiers, au début semble t'il, d'après une étude du cabinet Gartner group (1998).

b. Outils de développement

➤ Langage de développement

Le langage standard utilisé par l'UAB-vie est le PHP en raison de sa puissance et de son caractère Open Source.

Créer en 1994 par RAMSUS Lederf, PHP signifiait à cette époque Personal Home Page. C'est en 1997 que PHP devient Hypertext Préprocessor grâce aux travaux de ZEEV Suraski et ANDI Gutmans qui récrivirent l'interpréteur de PHP pour donner la version 3.

PHP a connu un succès fulgurant dans le domaine de la programmation web surtout auprès de la communauté GNU Linux en raison de sa gratuité et de son dynamisme. En effet PHP est passé du langage interprété (PHP3) au langage compilé (PHP4).

PHP est un langage intégré au code HTML et exécuté coté serveur. Il dérive du langage C et du PERL dont il reprend la syntaxe. Il est extensible grâce à de nombreux modules et son code source est ouvert.

La différence entre PHP et JAVASCRIPT réside dans le fait que PHP est exécuté coté serveur c'est-à-dire sur le serveur qui héberge l'application web tandis que JAVASCRIPT est exécuté sur le poste du client.

➤ La gestion des données

La gestion du contenu manipule beaucoup de données qui nécessitent d'être centralisées afin de faciliter la manipulation. L'outil permettant de centraliser et de manipuler les données est le SGBD ou Système de Gestion de Bases de Données qui peut être relationnel ou objet. Les objectifs visés par les SGBD sont les suivants :

- l'indépendance physique et logique des données.
- l'efficacité des accès aux données
- l'administration centralisée des données
- la cohérence des données
- la sécurité des données
- la résistance aux pannes
- la manipulation des données par des non informaticiens.

Il existe plusieurs types de SGBD mais les plus répandus sont ORACLE, POSTGRESQL, MYSQL.

SGBD : MYSQL

Une base de données étant un ensemble de données organisées et accessible par plusieurs utilisateurs travaillant en parallèle, un SGBD est une interface entre la base de données et les utilisateurs.

Le SGBD MySQL est un SGBD relationnel (SGBDR) qui dérive directement de SQL qui est un langage de requête vers les bases de données exploitant le modèle relationnel. Il reprend la syntaxe de SQL mais des opérations telles que les sélections

imbriquées et les clés étrangères n'apparaissent pas dans MySQL. MySQL est très souvent utilisé avec le langage de création de page web dynamique PHP. Comme tous les SGBD MySQL dispose de beaucoup de fonctionnalités permettant de répondre aux exigences minima des utilisateurs.

V. Diagnostic du système existant

Le système actuel a le seul avantage d'être traité par des hommes donc qui peuvent lors du traitement détecter des erreurs qui n'ont pas été signalées par l'assuré et qui sont profitables pour l'UAB-vie.

Mais il faut noter aussi que le système de gestion actuel du contentieux souffre de beaucoup de lacunes. Nous avons pu relever les points faibles suivants :

- **la gestion des données** : il n'existe pas une politique efficace de la gestion des données nécessaires pour le traitement du contentieux. En effet les données sont dispersées un peu partout dans le service rendant la recherche des informations difficile.
- **la politique d'attribution des numéros de pièces comptables** : la politique d'attribution actuelle n'optimise pas le travail. En effet avec la politique actuelle plusieurs mois peuvent avoir le même numéro de pièce comptable. Ceci n'est pas une bonne règle de gestion car elle ne facilite pas le pointage.
- **le nombre de personne chargé du contentieux** : le nombre de personne effectuant le travail du contentieux est faible par rapport au nombre de client de l'UAB-vie. Il faut rappeler que l'UAB-vie occupe la première place sur le plan nationale dans le domaine de l'assurance-vie. Le nombre réduit de personne fait que le traitement du contentieux évolue très lentement.

C. SCENARII

I. Objectifs

L'objectif de cette étape est de présenter divers scénarii de mise en œuvre du futur système.

Au vu des contraintes financières et organisationnelles de l'UAB-vie nous proposons trois scénarii pour l'automatisation du contentieux. Chaque approche présente l'architecture réseau, le matériel et logiciels nécessaires pour la mise en œuvre ainsi que le coût de réalisation total du projet.

Une étude comparative est faite afin de dégager clairement les avantages et les inconvénients de chaque approche.

II. Scénario d'exception

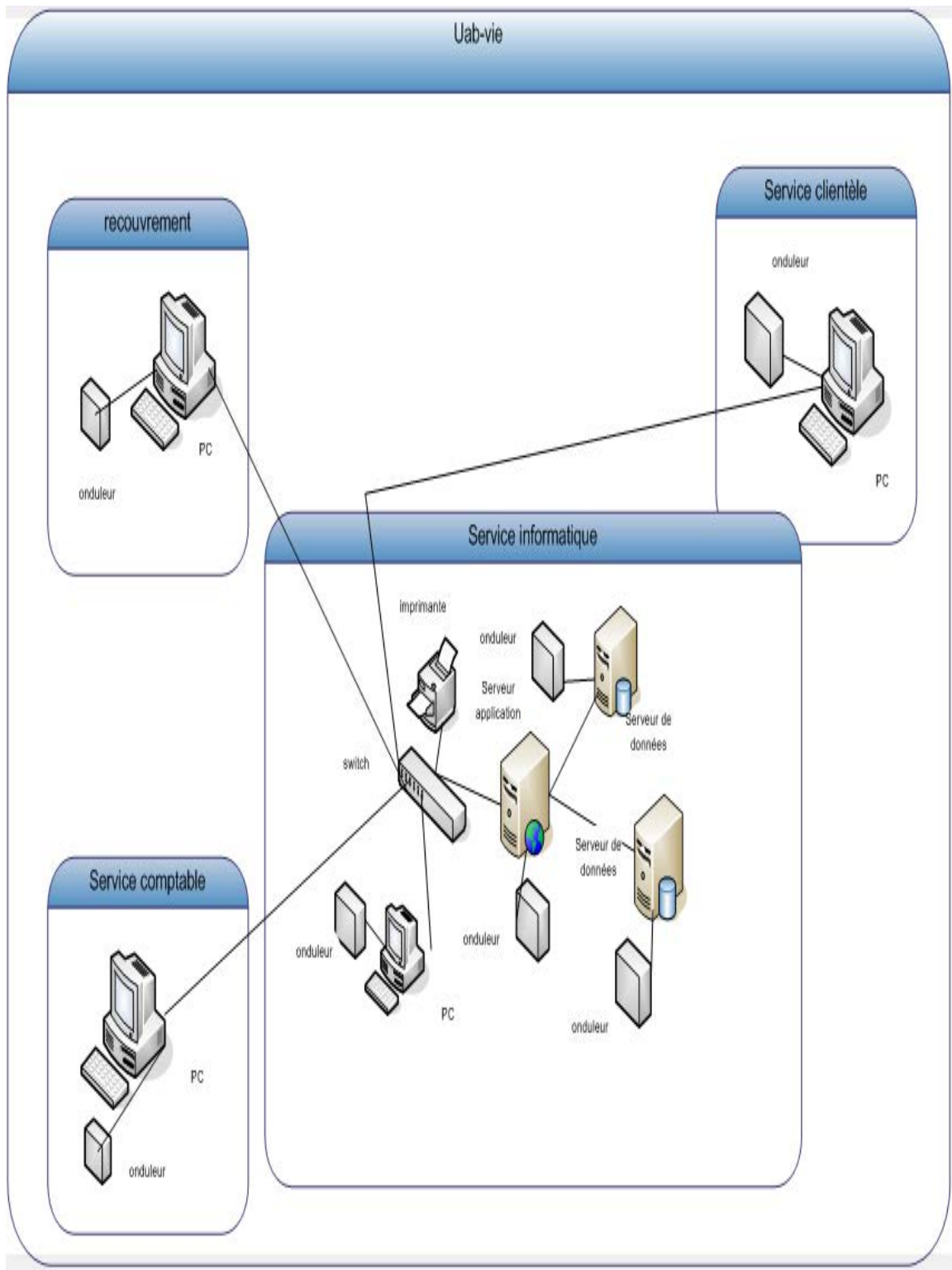
Ce scénario est une version automatisée du système actuel. L'application est de type Client/Serveur et développée en php. Elle est hébergée par un serveur web apache situé au service informatique et est accessible à travers le réseau local. Ce scénario permet un suivi de l'ensemble des activités du recouvrement et contentieux à savoir :

- **les réclamations** : l'application permet d'automatiser le traitement du pointage et de rechercher directement des mois concernés par un litige. Le SCV pourra, à travers l'application, transmettre les différentes demandes au recouvrement et contentieux et connaître le statut de ces demandes.
- **suivi des entrées de chèques et des virements** : l'ensemble des chèques et des ordres de virements sont enregistrés par le comptable. Ces enregistrements sont accessibles par le recouvrement et contentieux leur permettant ainsi de connaître les partenaires qui sont à jour.
- **les modifications** : l'application permettra d'avoir un suivi des ordres de modifications de prime, d'annulation de contrat. Il permettra de gérer également les ordres de prélèvement pour les nouvelles productions.

L'application accède à deux bases de données. La première base est celle existante et contenant les encaissements des assurés. La seconde sera créée pour le suivi des différents documents.

II.1 Représentation du flux d'informations

Uab-vie



II.2 Matériels nécessaires

Désignation	Caractéristiques	Quantité	Prix unitaire	Prix total
Serveur	Processeur : 2,4 GHz, 65 watts, 4 Mo de mémoire cache. Mémoire de masse : 587 Go	01	1600000	1600000
Micro-ordinateur	Pentium IV	05	Existant	
Onduleur	Autonomie d'au moins une heure	08	144534	1156272
Switch	24 ports	01	149231	149231
imprimante	HP laser 4200	01	Existant	
Serveur web	Apache	01	Gratuit	
Système d'exploitation	Windows XP/Linux	01	Existant	
Anti-virus	Avast	01	Existant	
Environnement de développement	Macromedia	01	Gratuit	
Sgbd	MySQL	01	Existant	
Total				2905503

II.3 Coût de développement : La méthode COCOMO simplifié

La méthode utilisée pour le calcul du coût de développement est COCOMO. C'est une méthode qui permet de calculer l'effort à consentir pour le développement d'un logiciel ainsi que le temps nécessaire en fonction de plusieurs paramètres.

Inventé en 1970 par **BARRY BOEHM**, COCOMO a connu une évolution au cours du temps. Il existe de nos jours, le modèle COCOMO II plus adapté à l'aspect réutilisation des composants.

- **Principe de COCOMO**

La méthode est divisée en trois modèles, qui permettent de faire une estimation en prenant en compte de plus en plus de paramètres :

- ✓ **le modèle de base** qui permet de déterminer l'effort et le temps nécessaires en fonction du nombre d'instructions que l'application doit contenir et la complexité de cette dernière. Une ventilation est également possible. La ventilation permet de calculer le temps de développement ainsi que l'effort à fournir pour chaque phase du cycle de développement.
- ✓ **le modèle intermédiaire** qui reprend l'effort et la durée du modèle de base mais en appliquant cette fois-ci des coefficients prenant en compte les facteurs de coût (compétence de l'équipe, complexité de l'environnement technique, etc.).
- ✓ **le modèle détaillé** qui reprend les résultats du modèle intermédiaire en affinant notamment les facteurs de coût en fonction de chaque phase du cycle de développement. Ce modèle n'est véritablement nécessaire que pour de très gros projets.

- **Complexité**

Il existe trois types de complexité :

S = application simple, n'ayant que peu de cas particuliers et de contraintes.

P = application intermédiaire mais plus complexe que les applications de type S.

E = application très complexité, que ce soit au niveau de leurs contraintes ou au niveau des données saisies.

En fonction de la complexité de l'application, on utilisera différents coefficients prenant en compte les différentes complexités et forcerment les différents efforts à fournir.

- **Modèle de base**

Nous avons choisit de détailler le modèle de base car nous allons effectuer nos calculs en fonction des principes de ce modèle.

- **Les formules**

S: Effort = $2,4 * KLS^{1,05}$ TDev = $2,5 * Effort^{0,38}$

P: Effort = $3 * KLS^{1,12}$ TDev = $2,5 * Effort^{0,35}$

E : Effort = $3,6 * KLS^{1,2}$ TDev = $2,5 * Effort^{0,32}$

Unité :

Effort en **HommeMois**.

TDev en **Mois**

KLS= Kilo Ligne Source représente le nombre de milliers d'instructions que contiendra l'application.

Le plus complexe est la détermination de nombre de KLS. A première vue, on peut se dire que c'est une chose impossible ou avec une très grande marge d'erreur. Cependant, pour être valable le modèle COCOMO ne doit être utilisé que lorsque la phase de conception est déjà bien avancée, de manière à avoir une idée assez précise du travail à réaliser. De plus l'expérience des développeurs est déterminante, car ils seront ainsi en mesure de s'approcher au plus près du bon nombre de KLS.

- **Calcul du coût du projet**

L'application à développer pour l'automatisation du contentieux S vu qu'il y'a beaucoup de contraintes et de cas à gérer.

Pour ce qui concerne le nombre de lignes du code, nous l'estimons à 5000 KLS .Ce qui donne le calcul suivant :

$$\text{EFFORT} = 2,4(5)^{1,05} = 2,4 * 5,41 = 13\text{HV}$$

$$\text{TDev} = 2,5 * (13)^{0,35} = 6,62 \text{ mois}$$

$$\text{NB pers} = 13 / 6,62 = 1,96 \sim 2 \text{ personnes}$$

$$\text{Coût financier} = 6,62 * 200000 = 2640000 \text{ FCFA}$$

II.4 Coût de la formation

Coût par personne	Nombre d'utilisateurs	Coût total
5000	8	40000 FCFA

II.5 Coût total du projet

désignation	coût	coût total
matériel	2905503	2905503
développement	2640000	2640000
formation	40000	40000
total		5585503

III. Scénario de compromis

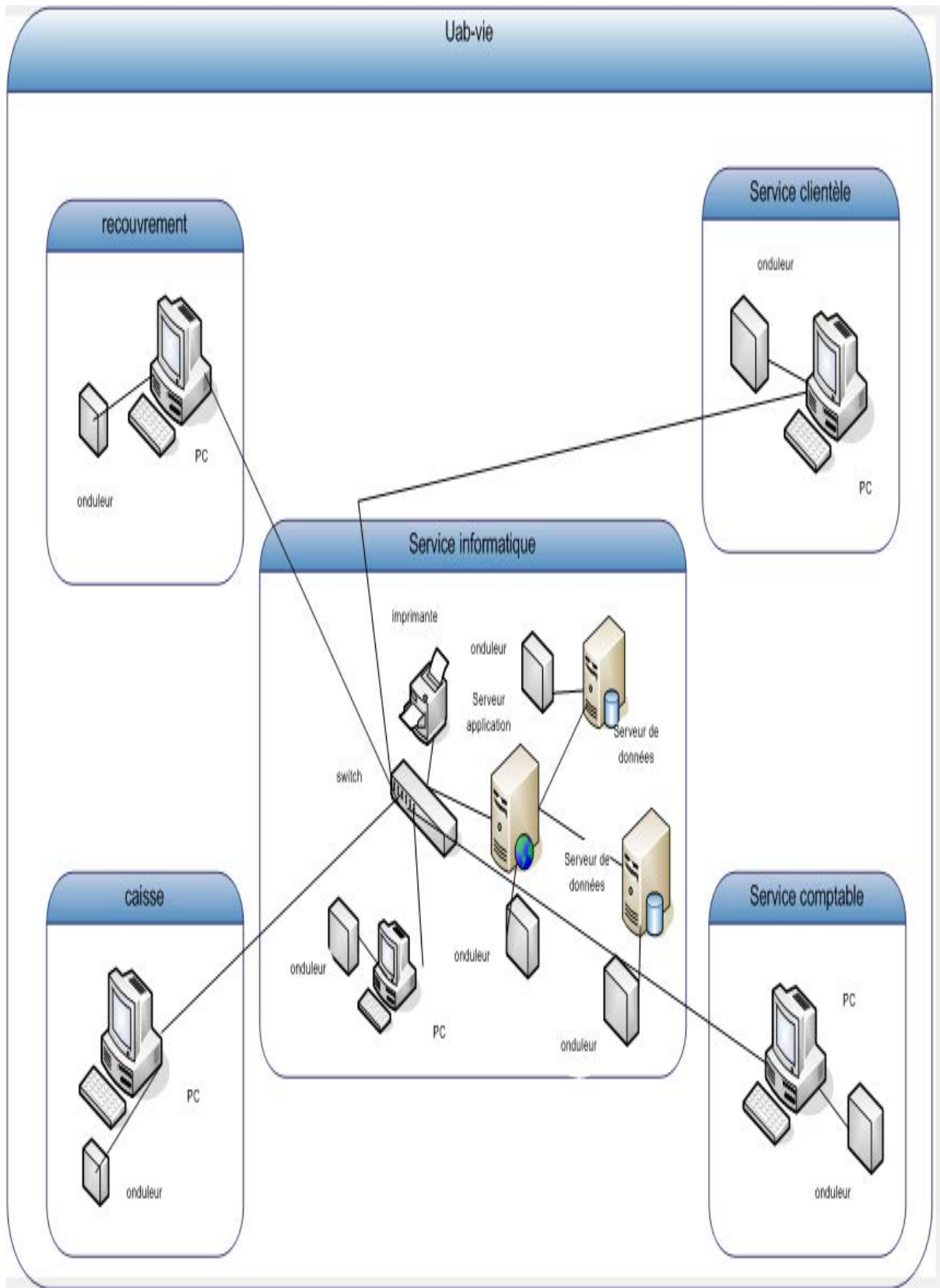
Ce scénario est presque identique au précédent à la différence que chaque service enregistre les documents qui lui parviennent. L'application est de type Client/Serveur et développée en php. Elle est accessible à travers le réseau local via un serveur web apache situé au service informatique.

Le pointage est automatisé et les corrections sont transmises à l'informatique pour la mise à jour. Le SVC pourra transmettre les différentes requêtes au RC et suivre l'évolution du traitement de ces requêtes.

Le comptable enregistre les entrées de chèques et les ordres de virements. La caissière est chargée de transmettre les cas de paiement par chèque au comptable pour l'attribution des numéros de pièces comptables. Ces différents enregistrements sont accessibles par le RC leur permettant ainsi de connaître les partenaires qui sont à jour. L'application permettra d'avoir un suivi des ordres de modification de prime, d'annulation de contrat et gérer les ordres de prélèvement pour les nouvelles productions. L'application accède également à deux bases de données : une base pour le pointage (existante) et une base pour le suivi des documents utilisés

III.1 Représentation du flux d'informations

Uab-vie



III.2 Les besoins matériels et logiciels

Désignation	Caractéristiques	Quantité	Prix unitaire	Prix total
Serveur de base de données	Processeur : 2,4 GHz, bus frontal 1066 MHz, 4 Mo de mémoire cache. Mémoire de masse : 587 Go	01	1600000	1600000
Micro-ordinateur	Pentium IV	05	Existant	
Onduleur	Autonomie d'au moins une heure	08	144534	1156272
Switch	24 ports	01	149231	149231
imprimante	HP laser 4200	01	Existant	
Serveur web	Apache	01	Gratuit	
Système d'exploitation	Windows XP/Linux	01	Existant	
Anti-virus	Avast	01	Existant	
Environnement de développement	Macromedia	01	Gratuit	
Sgbd	MySQL	01	Existant	
Total				2905503

III.3 Coût de développement

Ce deuxième scénario nécessite un effort de développement plus élevé que le premier scénario car la base de données est entièrement conçue. Pour cela nous estimons à 5100 le nombre de ligne du code. En suivant les formules du modèle de base de la méthode COCOMO détaillée plus haut nous obtenons les calculs suivants :

Effort= $2,4 * (5100 / 1000)^{1,05} = 13,27$ HM

- **TDev**= $2,5 * (13,27)^{0,38} = 6,67$ mois
- **NBpersonne**= Effort / TDev= $13,27 / 6,67 = 1,98 \sim 2$ personnes
- **Coût de développement**= $6,67 * 200000 * 2 = 2668000$ FCFA

III.4 Coût de formation

Coût par personne	Nombre d'utilisateur	Coût total
5000	8	40000 FCFA

III.5 Coût total de réalisation

Désignation	Coût (FCFA)
Matériels	2905503
Développement	2668000
Formation	40000
Total	5613503

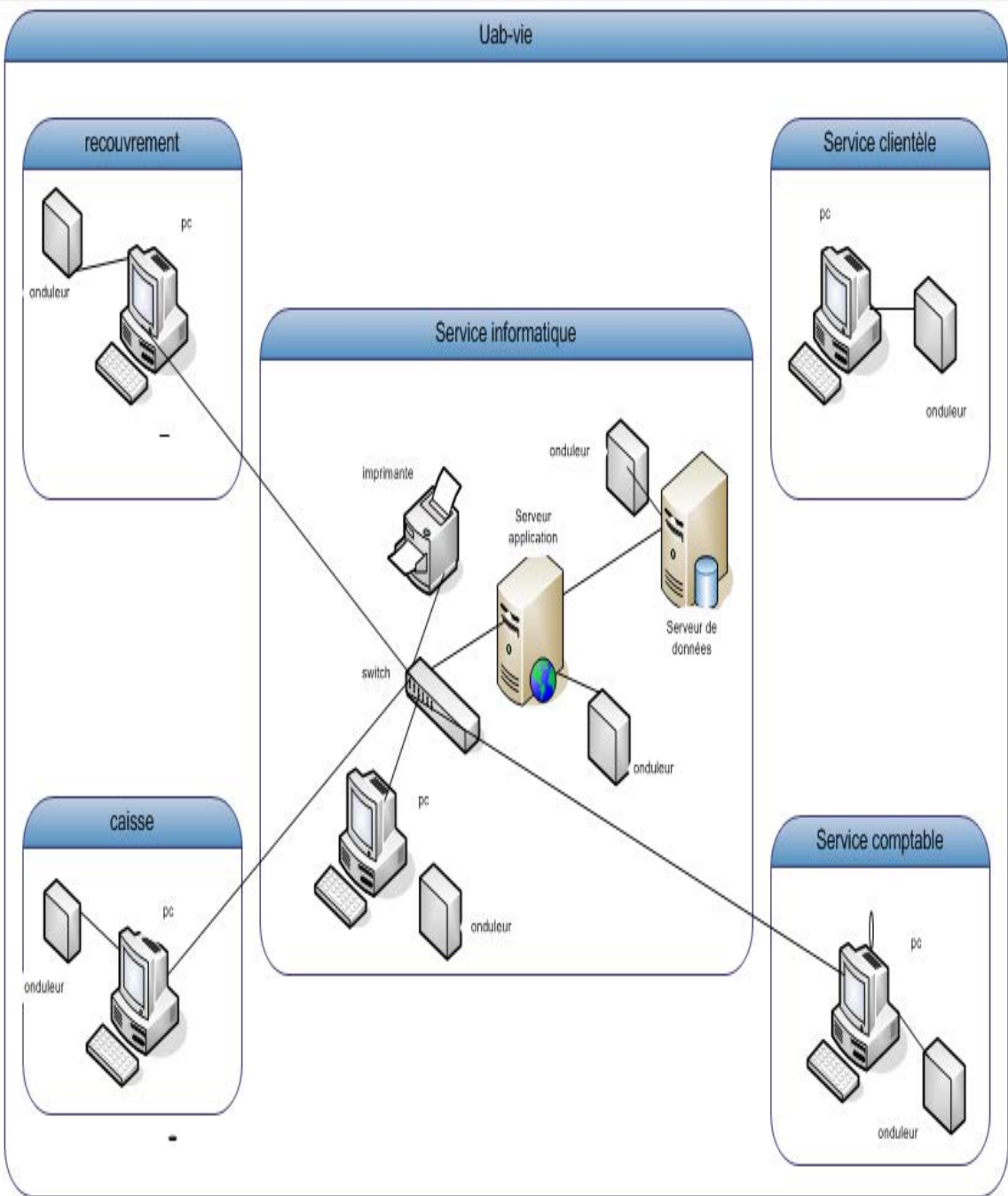
Tableau du coût total du projet

IV. Scénario idéal

Ce scénario est plus indépendant que les autres scénarii en ce sens qu'il utilise une base de données autonome et différente de la base de données existante. L'application est accessible à travers le réseau local via un serveur apache situé au service informatique. Le groupe d'utilisateurs effectue les mêmes opérations que dans le scénario précédent. La saisie des encaissements se fera à travers l'application. Cela permet un contrôle rigoureux de la saisie étant donné que beaucoup d'erreurs interviennent lors de la saisie.

IV.1. Représentation du flux d'informations

Uab-vie



IV.2 Matériels nécessaires

Désignation	Caractéristiques	Quantité	Prix unitaire	Prix total
Serveur de base de données	Processeur : 2,4 GHz, 65 watts, 4 Mo de mémoire cache. Mémoire de masse : 587 Go	01	1600000	1600000
Micro-ordinateur	Pentium IV	05	Existant	
Onduleur	Autonomie d'une heure au moins	08	144534	1156272
Switch	24 ports	01	149231	149231
imprimante	HP laser 4200	01	Existant	
Serveur d'application	Apache	01	Gratuit	
Système d'exploitation	Windows XP/Linux	01	Existant	
Anti-virus	Avast	01	Existant	
Environnement de développement	Macromedia	01	Gratuit	
Sgbdr	MySQL	01	Existant	
Total				2905503

IV.3 Coût du développement

Ce scénario traite le problème dans sa globalité et la base de données est entièrement conçue. Pour cela nous estimons à 6000 le nombre de ligne de code. Toujours en utilisant la méthode COCOMO nous obtenons les calculs suivants :

Effort = $2,4 * (6000/1000)^{1,05} = 15,74$ HM

TDev = $2,5 * (15,74)^{0,38} = 7,12$ mois

NBpersonne = $15,74 / 7,12 = 2,20 \sim 2$ personnes

Coût de développement = $7,12 * 200000 * 2 = 2848000$ FCFA

V.4 Coût de formation

Coût par personne	Nombre d'utilisateur	Coût total
5000	8	40000 FCFA

Tableau des coûts de formation

IV.5 Coût total du projet

Désignation	Coût (FCFA)
Matériels	2905503
Développement	2848000
Formation	40000
Total	5793503

Tableau du coût total de formation

V. Comparaisons des scénarii

Cette comparaison à pour but de faire percevoir les avantages et les inconvénients de chaque scénario.

V.1 Scénario d'exception

➤ Avantages

- sécurisé
- coût de réalisation raisonnable
- mise en œuvre facile
- répond aux besoins des utilisateurs
- facilite l'accès à l'information
- maintenance facile

➤ Inconvénients

- surcharge pour le comptable
- risque de perte d'informations
- risque d'erreur pendant la saisie des encaissements ou pendant la correction

V.2 Scénario de compromis

➤ Avantages

- Système sécurisé
- gain en temps de traitement
- coût de mise en œuvre moyen
- répartition des tâches
- mise en œuvre facile
- répond aux besoins de l'entreprise
- maintenance facile
- évolutif

➤ Inconvénients

- risque d'erreur pendant la saisie des encaissements ou pendant la correction

V.3 Scénario extrémiste

➤ Avantages

- gestion des données cohérente
- gain en temps de traitement
- répond aux besoins de l'entreprise
- maintenance facile

➤ Inconvénients

- mise en œuvre difficile
- coût de réalisation élevé
- surcharge pour l'informaticien

V.4 Récapitulatif de l'étude comparative des scénarii

scénario	Niveau de sécurité	Traitement	Intégrité des données	Coût de réalisation	Mise en oeuvre
Scénario d'exception	Elevé (9)	Moyen (4)	Moyen (5)	Faible (9)	Facile (9)
Scénario de compromis	Elevé (9)	Efficace (9)	Moyen (5)	Moyen (8)	Facile (9)
Scénario Idéal	Elevé (9)	Efficace (9)	Elevé (9)	Elevé (7)	Difficile (4)

***Moyenne** : la moyenne s'obtient en faisant la somme totale sur le nombre de critère.

- scénario d'exception : 7.2
- scénario de compromis : 8
- scénario Idéal : 7.6

D. SCENARIO RETENU

L'un des problèmes révélés lors de l'étude de l'existant est la perte des informations due au fait que ces dernières sont dispersées un peu partout dans l'entreprise. Pour palier à ce problème l'UAB-vie veut, dans le futur, centraliser la réception des paiements. En effet, dans le futur tous les paiements se feront à la caisse.

Vu l'ensemble de ces contraintes l'UAB-vie a adopté le scénario de compromis car il répond parfaitement à la future politique qui sera mise en place. En effet, avec ce scénario les chèques sont enregistrés par le comptable et la caissière.

Ce scénario a été retenu aussi parce que sa mise en oeuvre est facile et son coût total de réalisation est moyen par rapport aux autres scénarii.

E. CONCEPTION DETAILLEE DE LA SOLUTION RETENUE

I. Reconfiguration du système existant

La reconfiguration du système consiste à faire un apport dans le but d'améliorer le système existant. Dans notre cas elle va consister à améliorer les cas d'utilisations existant.

Ainsi en plus des cas d'utilisations que nous avons définis dans la phase d'Inception, nous proposons les cas d'utilisations repartis en fonction du domaine

I.1 Gestion des utilisateurs

- **authentification** : Ce cas d'utilisation concerne tous les acteurs car désormais pour faire une quelconque opération il faut s'authentifier. Ce cas d'utilisation se justifie par le fait que tous les utilisateurs n'effectuent pas les mêmes opérations. En fonction de ses droits l'utilisateur aura accès à des fonctions bien précises.
- **administrer** : l'administration est faite par une personne désignée pour cette tâche. Elle pourra grâce à ce cas d'utilisation créer ou supprimer des utilisateurs.

I.2 Gestion du contentieux

- **consulter demande** : désormais les transmissions de demandes se font à travers l'application. Le cas d'utilisation « consulter demande » permet au chargé du contentieux de voir les demandes qui ont été envoyées par le SCV.
- **suivre demande** : ce cas d'utilisation a été créé pour permettre au SCV de suivre le niveau d'évolution d'une demande qu'il a envoyée. Ainsi il pourra suivre l'assuré sur le niveau de traitement de sa requête.

- **changer statut demande** : permet de préciser le niveau de traitement d'une demande.
- **transmettre correction** : après le traitement d'une demande de réclamation, s'il y a des corrections à porter, ce cas d'utilisation permettra de transmettre le rapport de correction à l'informaticien.
- **consulter correction** : ce cas d'utilisation permettra à l'informaticien de consulter le rapport de correction envoyé par le chargé de contentieux afin de pouvoir faire les mises à jour.

Le cas d'utilisation « gérer contentieux » est subdivisé en plusieurs cas d'utilisation qui sont :

- **vérifier numéros manquants** : permet de recenser les numéros de pièces comptables qui ne figurent pas sur les encaissements de l'assuré.
- **vérifier montants** : permet de savoir si la prime enregistrée lors des encaissements correspond à la prime normale.
- **vérifier paiement par chèque** : permet de voir si les numéros manquants correspondent à des paiements par chèque.
- **vérifier paiement en espèce** : permet de voir si les numéros manquants correspondent à des paiements en espèce.
- **vérifier autres contrats** : permet de vérifier si les numéros manquants n'ont pas été saisi dans d'autres contrats et si tel est le cas il faut vérifier les montants correspondants à ces numéros.

I.3 Gestion des informations

- **enregistrer documents** : Ce cas d'utilisation permettra au chargé de contentieux d'enregistrer les différents documents qui lui sont envoyés et d'avoir un suivi des différentes demandes qu'il traitera.
- **rechercher document** : à travers l'application, le chargé de contentieux pourra rechercher des documents qui ont été enregistrés.

II. Diagrammes

II.1 Diagramme de cas d'utilisation

1. Acteurs :

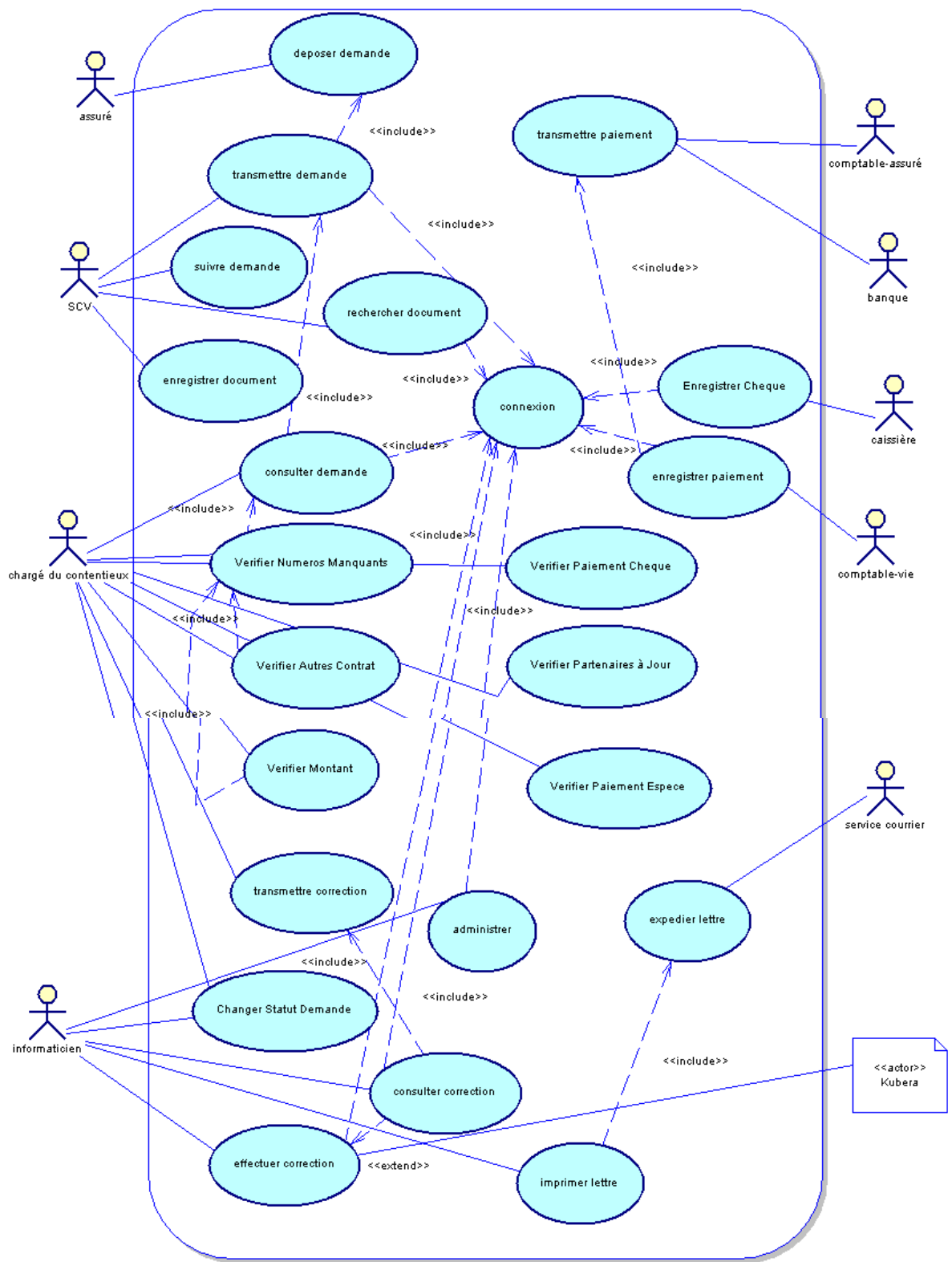
Assuré, le SCV, la caissière, Comptable-vie, Comptable-assuré, banque, chargé du contentieux, informaticien, SC, kubera.

2. Cas d'utilisation par acteur

- **assuré** : déposer demande
- **SVC** : transmettre demande, suivre demande, rechercher document
- **caissière** : enregistrer paiement en espèce ou par chèque.
- **comptable-vie** : enregistrer chèque ou ordre de virement
- **comptable-assuré** : transmettre chèque et listing
- **banque** : transmettre ordre de virement.

- **chargé du contentieux** : vérifier numéro manquant, vérifier montant, vérifier autres contrats, vérifier paiement en espèces, vérifier paiement par chèques, vérifier partenaires à jour, changer statut demande, transmettre correction, consulter demande, enregistrer document, rechercher document.
- **informaticien** : effectuer correction, administrer, consulter correction, imprimer lettre, changer statut demande.
- **service courrier** : expédier lettre.
- **kubera** : effectuer correction.

3. représentation des cas d'utilisation



4. description textuelle des cas d'utilisation

CU1 : Cas d'utilisation : déposer demande

Résumé : permet à l'assuré d'entamer une procédure de réclamation de nature différente.

Acteurs : assuré, SCV, chargé du contentieux

Préconditions : avoir établi au moins un contrat avec l'UAB-vie

Scénario nominal :

1. L'assuré rédige la demande
2. L'assuré transmet sa demande au commercial

Scénario alternatif :

1. Demande incorrecte : il faut répartir au point 1 du scénario nominal

CU2 : Cas d'utilisation : consulter demande

Résumé : permet au chargé du contentieux de consulter les demandes envoyées par le service clientèle.

Acteurs : chargé du contentieux, système.

Préconditions : il y a des demandes en instance.

Scénario nominal :

1. le chargé du contentieux demande à se connecter
2. le système demande le login et le mot de passe
3. il fournit le login et le mot de passe
4. le système affiche l'interface correspondant
5. il clique sur consulter
6. le système affiche les demandes en instance
7. le système propose de faire le pointage

Scénario alternatif :

1. **Le login ou le mot de passe est incorrecte** : dans ce cas il faut répartir au point 2 du scénario nominal
2. Il n'y a pas de demande en instance : il faut alors quitter l'application

Scénario d'exception

1. **Connexion impossible** : rétablir le réseau local
2. **Le système est en panne** : effectuer la maintenance

CU3 : Cas d'utilisation : transmettre correction

Résumé : permet, au cas où il y a des corrections à faire, de transmettre les corrections à faire à l'informaticien.

Acteurs : chargé du contentieux, informaticien, système

Préconditions : il y a des corrections à faire

Scénario nominal

1. Le chargé du contentieux clique sur « transmettre rapport ».
2. Le système transmet le rapport

Scénario d'exception

1. Le système est en panne : faire la maintenance

CU4 : Cas d'utilisation : consulter correction

Résumé : permet à l'informaticien de consulter les rapports de corrections envoyés par le chargé du contentieux

Acteurs : Informaticien, Système.

Préconditions : Le système est disponible

Scénario nominal

1. L'informaticien demande à se connecter
2. Le système demande le login et le mot de passe
3. L'informaticien s'authentifie
4. Le système fournit les fonctionnalités demandées
5. L'informaticien clique sur consulter
6. Le système affiche les corrections en instance

Scénario alternatif

1. **Login et mot de passe incorrect** : Retour au point 2 du scénario nominal
2. **Il n'y a pas de correction en instance** : l'informaticien se deconnecte

Scénario d'exception

1. **Le système est indisponible** : vérifier le réseau

CU5 : Cas d'utilisation : effectuer corrections

Résumé : ce cas d'utilisation permet à l'informaticien de corriger les erreurs détectées pendant la gestion du contentieux.

Acteurs : informaticien, Kubera

Préconditions : il y a des corrections à faire

Scénario nominal :

1. l'informaticien demande à se connecter
2. Kubera demande le login et le mot de passe
3. l'informaticien fournit le login et le mot de passe
4. Kubera lui présente les fonctionnalités nécessaires
5. l'informaticien fait la mise à jour
6. l'informaticien se déconnecte

Scénario alternatif :

1 Login ou mot de passe incorrecte : dans ce cas il faut repartir au point 3 du scénario nominal.

CU6 : Cas d'utilisation : imprimer lettre

Résumé : permet à l'assuré d'imprimer les lettres des assurés

Acteurs : Informaticien, Kubera

Scénario nominal

1. L'informaticien demande à se connecter
2. Kubera demande à l'informaticien de s'authentifier
3. L'informaticien s'authentifie
4. Kubera fournit les fonctionnalités
5. L'informaticien lance l'impression
6. Le système exécute

Scénario alternatif

1. **Login et mot de passe incorrecte** : retour au point 2 du scénario nominal

Scénario d'exception

2. **Kubera indisponible** : établir la connexion

CU7 : Cas d'utilisation : administrer

Résumé : permet à l'informaticien d'effectuer les opérations suivantes : ajouter un utilisateur, supprimer un utilisateur.

Acteurs : Informaticien, Système, Utilisateur.

Préconditions : il y'a un utilisateur à supprimer ou à ajouter

Scénario nominal

Ajouter un utilisateur

1. l'informaticien demande à se connecter
2. le système demande le login et le mot de passe
3. l'informaticien fournit le login et le mot de passe
4. le système affiche les outils d'administration
5. l'informaticien choisit « ajouter »
6. le système demande les informations nécessaires
7. l'informaticien fournit les informations nécessaires
8. l'informaticien valide les informations
9. le système ajoute l'utilisateur

Supprimer un utilisateur

1. l'informaticien demande à se connecter
2. le système demande le login et le mot de passe
3. l'informaticien fournit le login et le mot de passe
4. le système affiche les outils d'administration
5. l'informaticien choisit « supprimer »

6. le système demande les informations nécessaires
7. l'informaticien fournit les informations
8. l'informaticien valide les informations
9. le système supprime l'utilisateur

Scénario alternatif

Login ou mot de passe incorrecte : revenir au point 2 du scénario nominal

Scénario d'exception

1. Système indisponible : rétablir le réseau local

CU8 : Cas d'utilisation : enregistrer versement

Résumé : permet au comptable de l'UAB-vie d'enregistrer les versements et d'attribuer des numéros de pièces comptables.

Acteurs : Comptable-vie, système.

Préconditions : il y a des versements à enregistrer.

Scénario nominal

1. le comptable demande à se connecter
2. le système demande le login et le mot de passe
3. le comptable donne le login et le mot de passe
4. le système affiche les fonctions
5. le comptable choisit « enregistrer »
6. le système affiche la fenêtre de saisie
7. le comptable saisie les informations
8. il valide
9. le système fait la mise à jour

Scénario alternatif

1. login ou mot de passe incorrecte : retour au point 2 du scénario nominal

2. listing incorrect : faire appel à la société ou la banque.

Scénario d'exception

1. système indisponible : rétablir le réseau local

CU9 : Cas d'utilisation : enregistrer paiement en espèce ou par chèque.

Résumé : permet à la caissière d'enregistrer les cas de paiement en espèce ou par chèque.

Acteurs : caissière, système.

Préconditions : il y a des versements à enregistrer

Scénario nominal

1. la caissière demande à se connecter
2. le système demande le login et le mot de passe
3. la caissière fournit le login et le mot de passe
4. le système affiche l'interface correspondant
5. la caissière choisit enregistrement
6. elle saisit les données et valide

7. le système fait la mise à jour

Scénario alternatif

1. login ou mot de passe incorrecte : retour au point 2 du scénario nominal

Scénario d'exception

1. système indisponible : rétablir le réseau

CU10 : Cas d'utilisation : transmettre paiement

Résumé : permet à la banque et à la comptabilité d'envoyer la liste des assurés ayant autorisés le paiement.

Acteurs : banque, comptabilité-assuré, comptabilité-vie.

Préconditions : les assurés ont autorisés le paiement.

Scénario nominal

1. la banque ou la comptabilité effectue le prélèvement
2. ils établissent la liste des assurés
3. ils envoient la liste à l'UAB-vie

Scénario alternatif

1. le compte n'est pas approvisionné : avertir l'assuré

CU11 : Cas d'utilisation : expédier lettre

Résumé : lorsque l'informaticien tire les lettres des assurés, le service courrier est chargé de faire parvenir les lettres aux destinataires.

Acteurs : service courrier, assuré.

Préconditions : il y a des lettres à envoyer

Scénario nominal

1. le responsable du service courrier vérifie les lettres
2. il enveloppe les lettres
3. il expédie les lettres

Scénario alternatif

1. l'adresse est incorrecte : dans ce cas il faut ramener la lettre au service informatique pour tirer à nouveau.

CU12 : Cas d'utilisation : connexion

Résumé : permet à tous les acteurs primaires d'avoir accès au système.

Acteurs : tous les acteurs du système, le système.

Préconditions : avoir un compte sur le système

Scénario nominal

1. l'utilisateur demande à se connecter
2. le système demande le login et le mot de passe
3. l'utilisateur fournit le login et le mot de passe
4. le système affiche l'interface correspondant à l'utilisateur

Scénario alternatif

1. **login ou mot de passe incorrecte** : retour au point 2 du scénario nominal.

Scénario d'exception

1. **Système indisponible** : rétablir le réseau
2. **le système est en panne** : faire la maintenance

CU13 : Cas d'utilisation : suivre demande

Résumé : permet au SVC de connaître le niveau d'évolution du traitement des demandes envoyées.

Acteurs : SCV, chargé du contentieux, informaticien

Préconditions : il y a des demandes qui ont été envoyées au chargé du contentieux.

Scénario nominal

1. l'agent demande à se connecter
2. le système demande le login et le mot de passe
3. l'agent fournit le login et le mot de passe
4. le système affiche l'interface
5. l'agent clique sur statut
6. le système demande la date de transmission
7. l'agent fournit la date et valide
8. le système affiche les demandes correspondantes
9. l'agent vérifie le statut des demandes

Scénario alternatif

1. **Le login ou le mot de passe incorrecte** : repartir au point 3 du scénario nominal

Scénario exceptionnel

1. Le système est indisponible : rétablir le réseau
2. **Le système est en panne** : effectuer la maintenance

CU14 : Cas d'utilisation : enregistrement document

Résumé : permet au RC SCV d'enregistrer ses documents

Acteurs : recouvrement et contentieux

Préconditions : le document à enregistrer ne comporte pas des erreurs

Scénario nominal

1. l'agent demande à se connecter
2. le système demande le login et le mot de passe
3. l'agent fourni le login et le mot de passe
4. le système affiche l'interface

5. l'agent choisi « enregistrement »
6. le système affiche la fenêtre d'enregistrement
7. l'agent saisi les références du document et valide
8. le système enregistre les références

Scénario alternatif

1. **Le login ou le mot de passe incorrecte** : repartir au point 3 du scénario nominal

Scénario exceptionnel

1. **Le système est indisponible** : rétablir le réseau
2. **Le système est en panne** : effectuer la maintenance

CU15 : Cas d'utilisation : rechercher document

Résumé : permet de rechercher un document enregistrer au préalable.

Acteurs : SCV, RC

Préconditions : le document intervient dans l'exécution d'une opération donnée.

Scénario nominal

1. l'agent demande à se connecter
2. le système demande le login et le mot de passe
3. l'agent fourni le login et le mot de passe
4. le système affiche l'interface
5. l'agent choisi «rechercher »
6. le système demande les informations nécessaires pour la recherche
7. l'agent fourni les informations et valide
8. le système affiche le résultat de la recherche

Scénario alternatif

1. **le login ou le mot de passe est incorrecte** : repartir au point 3 du scénario nominal
2. les informations fournies sont incorrectes : repartir au point 6 du scénario nominal

CU16 : Cas d'utilisation : Vérifier numéros manquants

Résumé : cette opération consiste à faire une intersection entre les numéros de pièces comptables figurants sur le référentiel et ceux figurants sur l'état des encaissements de l'assuré. Les numéros qui figureront sur le référentiel et qui ne figurent pas sur la situation des encaissements de l'assuré feront l'objet d'autres vérifications. De même, les numéros qui figurent chez l'assuré et qui ne figurent pas sur le référentiel seront vérifiés parmi les numéros de cas de paiement par espèce ou par chèque.

Acteurs : chargé du contentieux, système.

Préconditions : disposer d'une demande de réclamation

Scénario nominal

1. cliquer sur pointage
2. le système affiche les différentes opérations concernant le pointage
3. choisir l'opération « vérifier numéros manquants »
4. le système affiche le résultat de l'opération « vérifier numéros manquants »
5. le système propose de vérifier si l'assuré a d'autres contrats afin d'y vérifier les numéros manquants

Scénario alternatif

1. **il n'existe pas de numéros manquants** : le système propose de vérifier les montants
Scénario d'exception
2. **la police de l'assuré est incorrecte** : recommencer le processus avec une police valide

CU17 : Cas d'utilisation : vérifier montants

Résumé : cette opération permet de savoir si les montants qui figurent sur l'état des encaissements de l'assuré sont exacts

Acteurs : chargé du contentieux, système

Préconditions : tous les numéros de pièce sont à jour

Scénario nominal :

1. cliquer sur « vérifier montant »
2. le système affiche le résultat de la vérification
3. le système propose d'envoyer un rapport de correction au service informatique

Scénario alternatif

1. **Tous les montants sont exacts** : le système propose de vérifier si les numéros concernés ne correspondent pas à des paiements par chèques ou en espèces

CU18 : Cas d'utilisation : vérifier autres contrats

Résumé : permet de savoir si l'assuré possède plusieurs contrats, si c'est le cas vérifier si les numéros manquants ne figurent pas dans ces contrats.

L'opération consiste à vérifier si les numéros manquants figurent sur les autres contrats de l'assuré. Si ces numéros figurent il faut alors si le montant est la somme des primes de l'ensemble des contrats. Si tel est le cas il faut donc répartir la somme en fonction de la prime de chacun des contrats.

Acteurs : chargé du contentieux, système

Préconditions : les numéros ne figurent pas sur le contrat concerné

Scénario nominal

1. choisir « vérifier autres contrats »
2. le système affiche le résultat de la vérification
3. le système propose d'envoyer un rapport de correction au service informatique.

Scénario alternatif

1. **les numéros ne figurent pas sur les autres contrats de l'assuré** : le système propose alors de vérifier si les numéros correspondants ne correspondent pas à des cas de paiement par chèque ou en espèce.
2. **il y a certains numéros qui figurent et certains qui ne figurent pas** : dans ce cas il faut effectuer le point 1 du scénario alternatif pour les numéros qui ne figurent pas non plus sur les autres contrats de l'assuré.

CU19 : Cas d'utilisation : vérifier paiement espèce

Résumé : permet de vérifier si les numéros ne figurants pas sur les encaissements de l'assuré ne sont pas des cas de paiement en espèce. Si les numéros figurent, il faut vérifier les montants correspondants

Acteurs : chargé du contentieux, système.

Préconditions : les numéros incriminés ne figurent ni sur les encaissements du contrat concerné ni sur les autres contrats de l'assuré.

Scénario nominal

1. cliquer sur « paiement en espèce »
2. le système affiche le résultat de la vérification.
3. si les numéros correspondent à des cas de paiement en espèce
4. le système propose de vérifier les montants.

Scénario alternatif

1. **les numéros ne correspondent pas à des paiements en espèce** : le système propose de vérifier si les numéros correspondent à des cas de paiement par chèque.

CU20 : Cas d'utilisation : vérifier paiement par chèque

Résumé : permet de voir si les numéros ne figurant pas sur les encaissements de l'assuré ne correspondent pas à des cas de paiement par chèque. Si les numéros figurent il faut vérifier les montants.

Acteurs : chargé du contentieux, système.

Préconditions : les numéros ne figurants pas sur les encaissements de l'assuré ne correspondent pas à des cas de paiement en espèce.

Scénario nominal

1. cliquer sur « paiement par chèque »
 2. le système affiche les résultats de la vérification
- si les numéros correspondent à des cas de paiement par chèque le système propose de vérifier les montants.

CU21 : Cas d'utilisation : changer statut demande

Résumé : permet de changer le statut d'une demande

Acteurs : chargé du contentieux, informaticien, SCV

Préconditions : la demande a été traitée au RC ou à l'informatique.

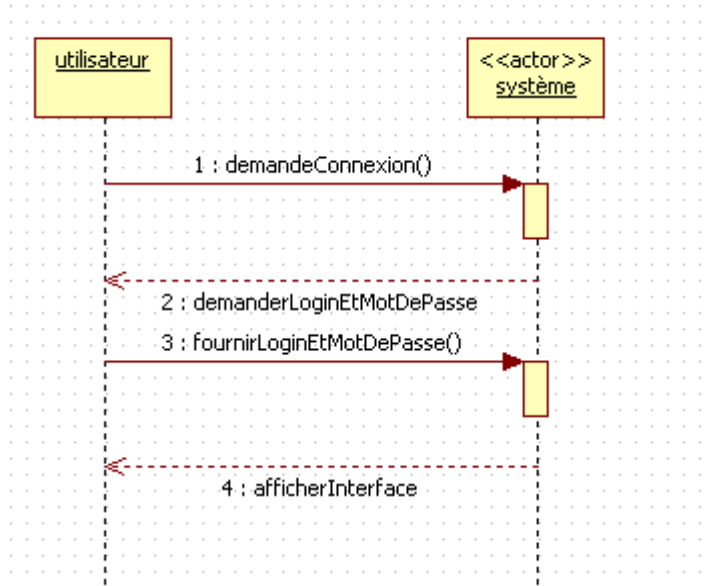
Scénario nominal

1. l'utilisateur clique sur « changer statut »
2. il saisit le statut et valide
3. le système fait la mise à jour

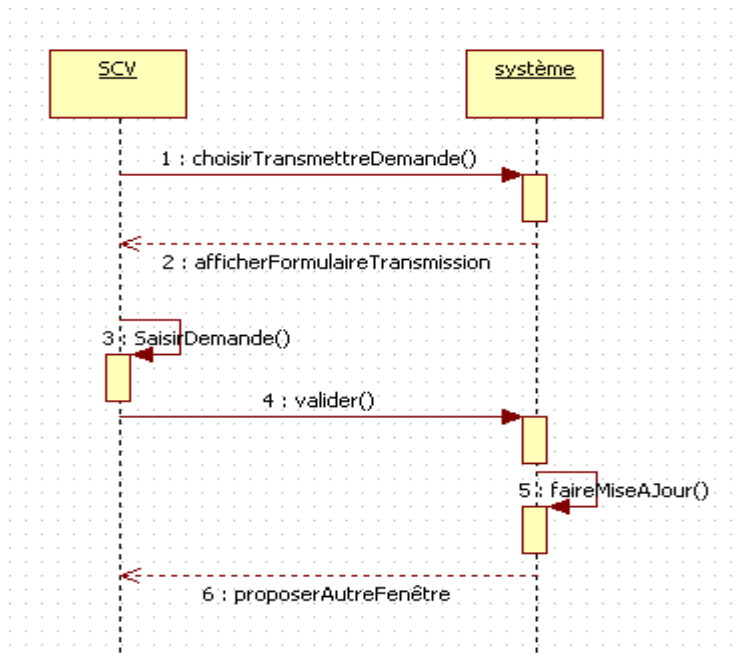
II.2 Diagramme de séquence

Nous avons choisi de décrire les cas d'utilisation les plus pertinents en occurrence ceux concernant le pointage et les procédures de connexion et de correction.

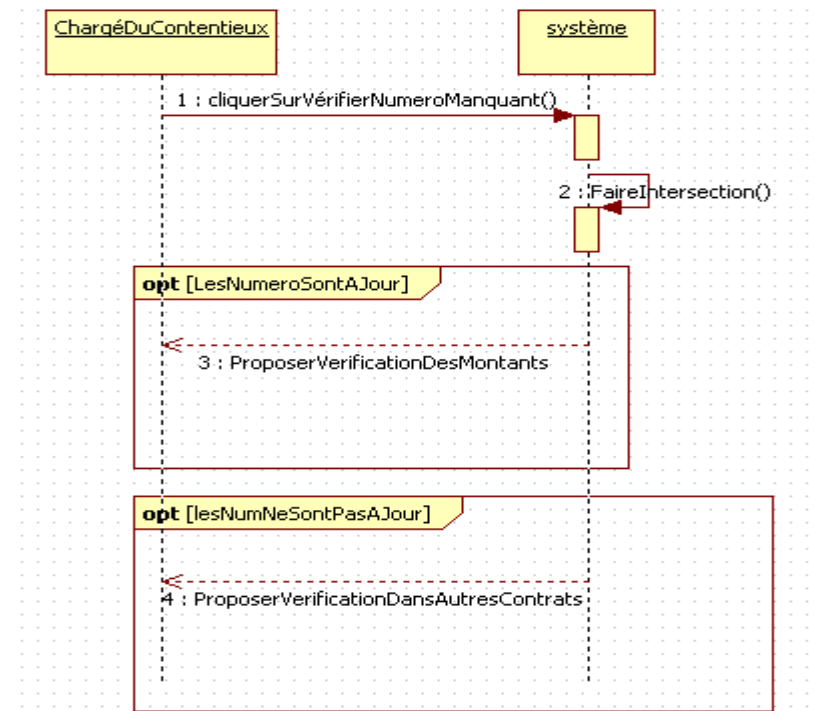
Connexion :



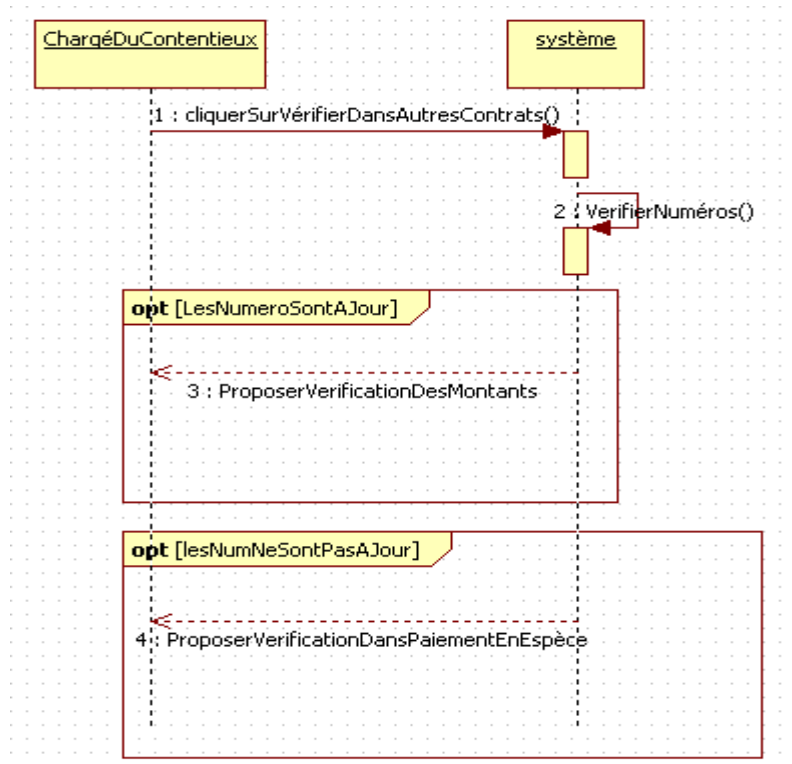
Transmettre demande :



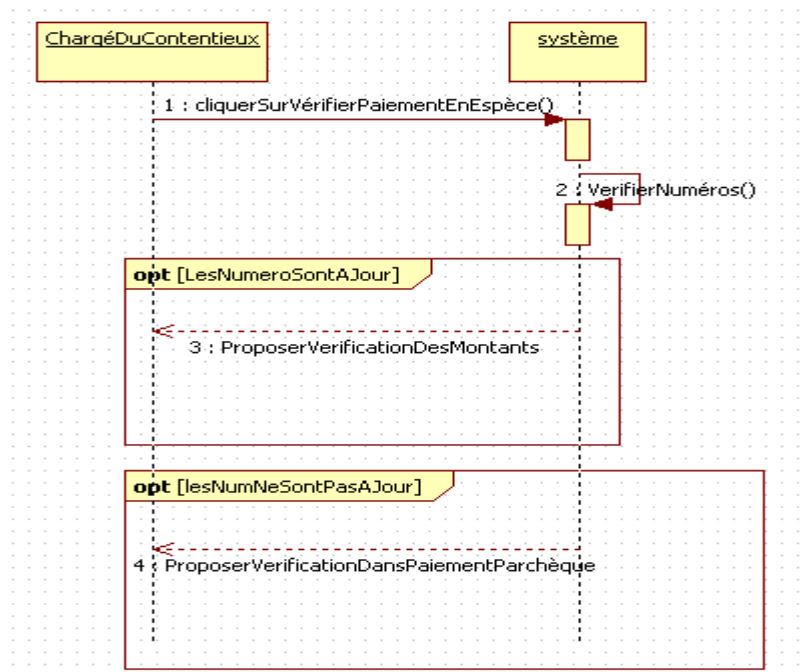
Vérifier numéros manquants :



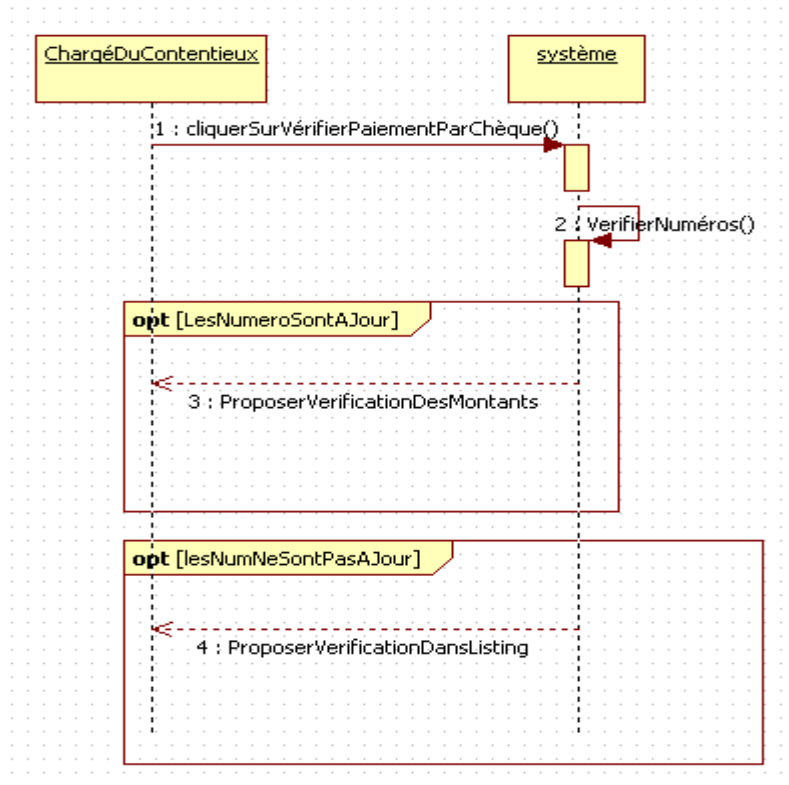
Vérifier dans autres contrats :



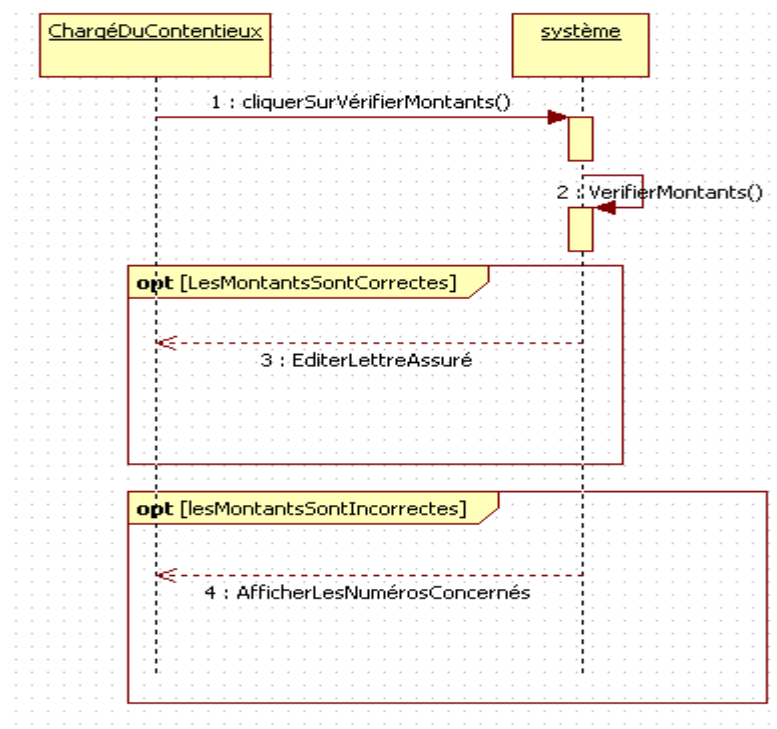
Vérifier paiement en Espèce :



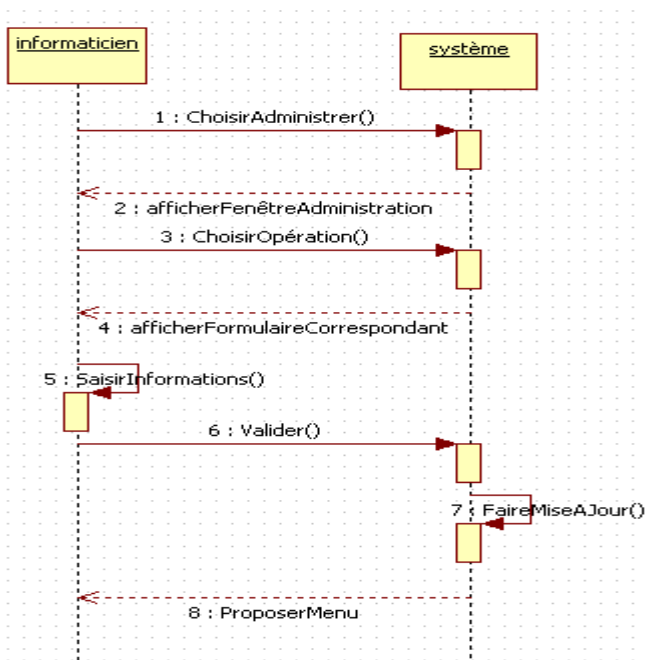
Vérifier dans paiement par chèque :



Vérifier montants :



Administrer :



II.3 Diagramme d'activités

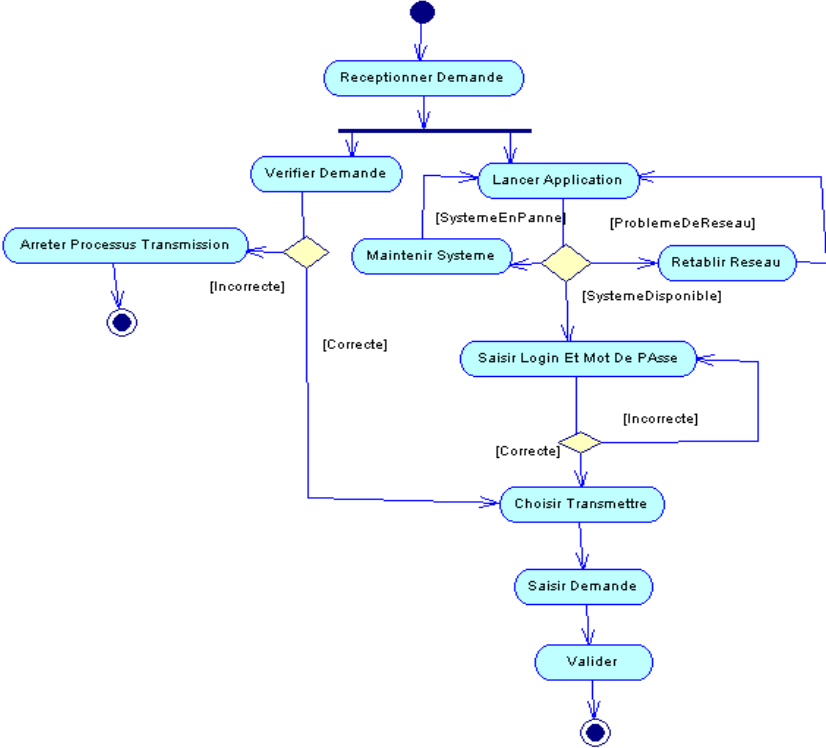
1. Définition et objectifs

Les diagrammes d'activités permettent de décrire les traitements. Ils permettent de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

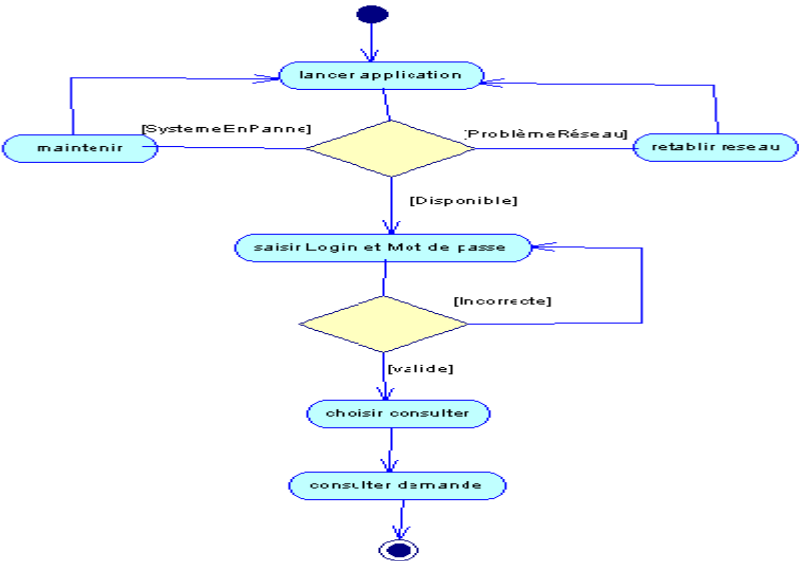
2. Représentation des diagrammes d'activités

Comme pour les diagrammes de séquences, nous avons choisi ici également de représenter les cas les plus pertinents.

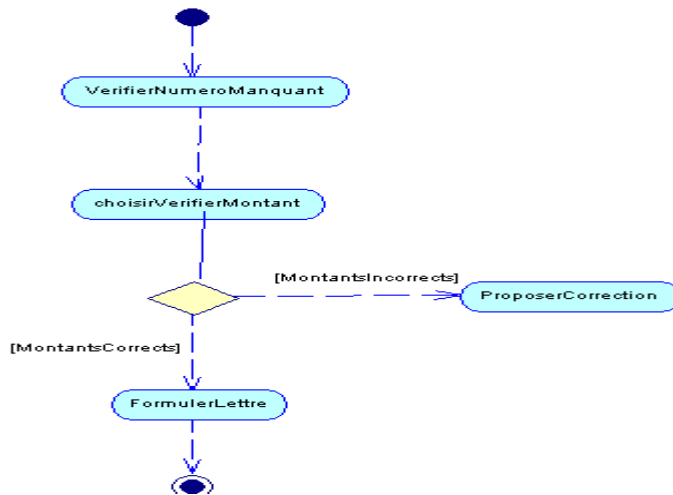
Transmettre demande :



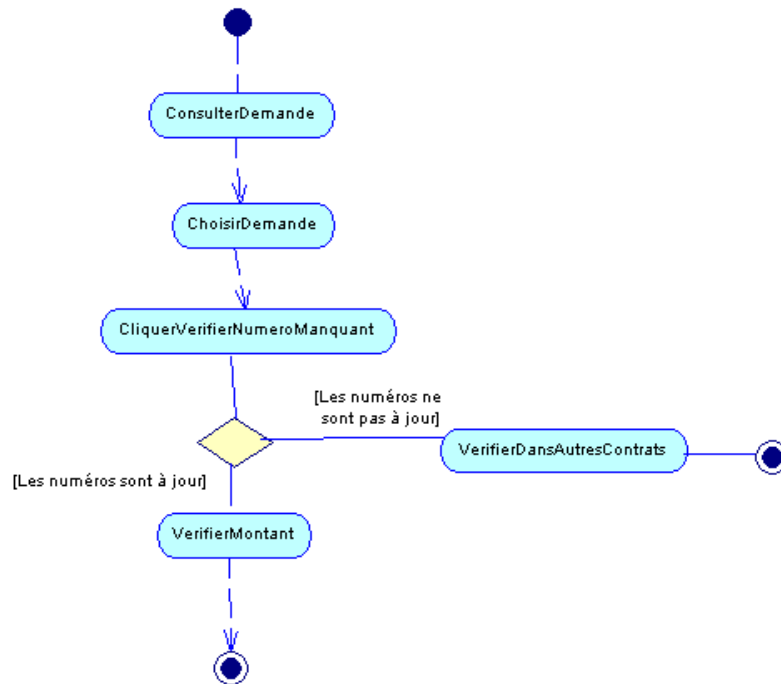
Consulter demande :



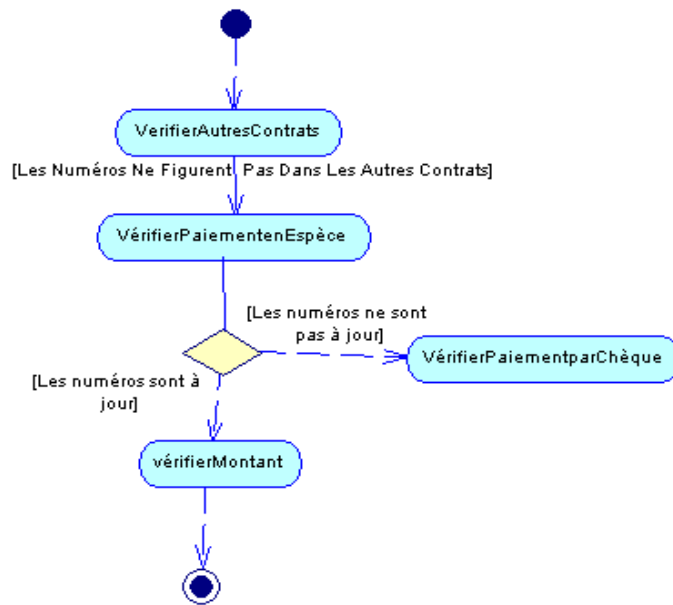
Vérifier numéros manquants :



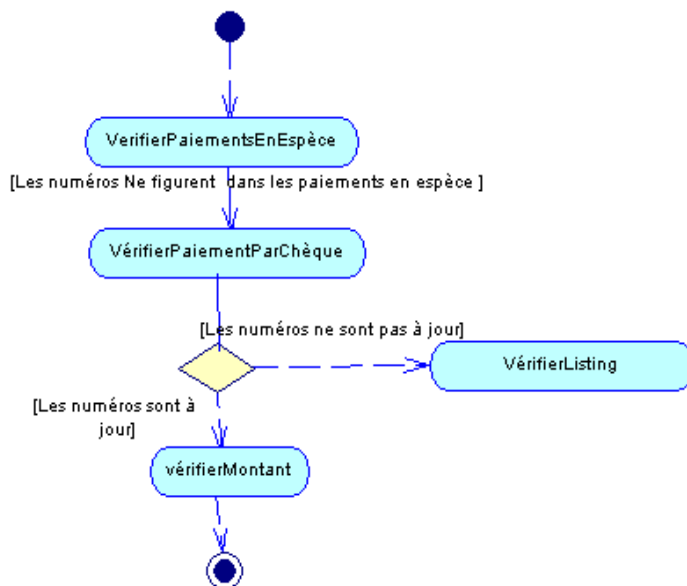
Vérifier les montants :



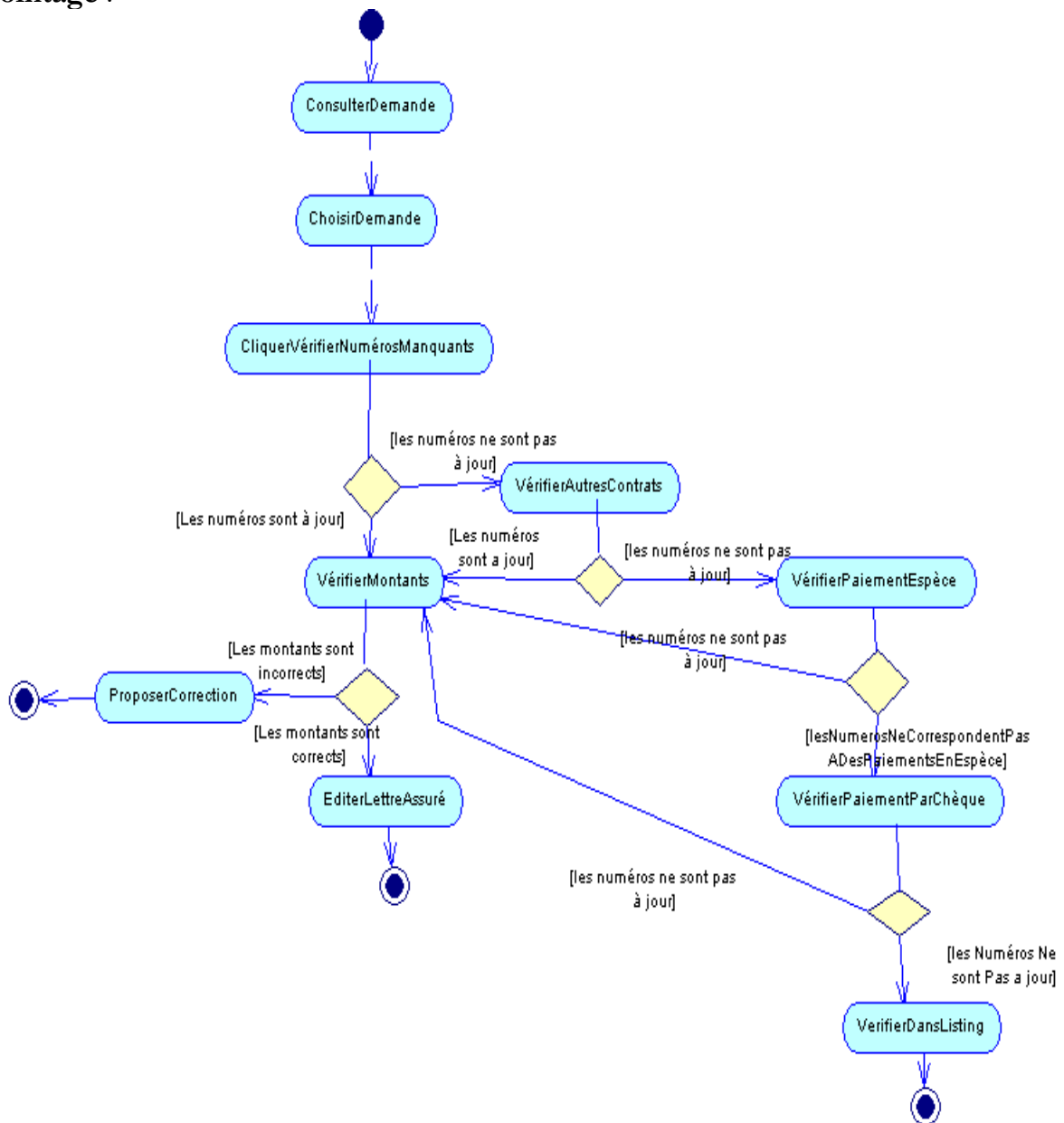
Vérifier les paiements en espèces :



Vérifier les paiements par chèques :



Pointage :

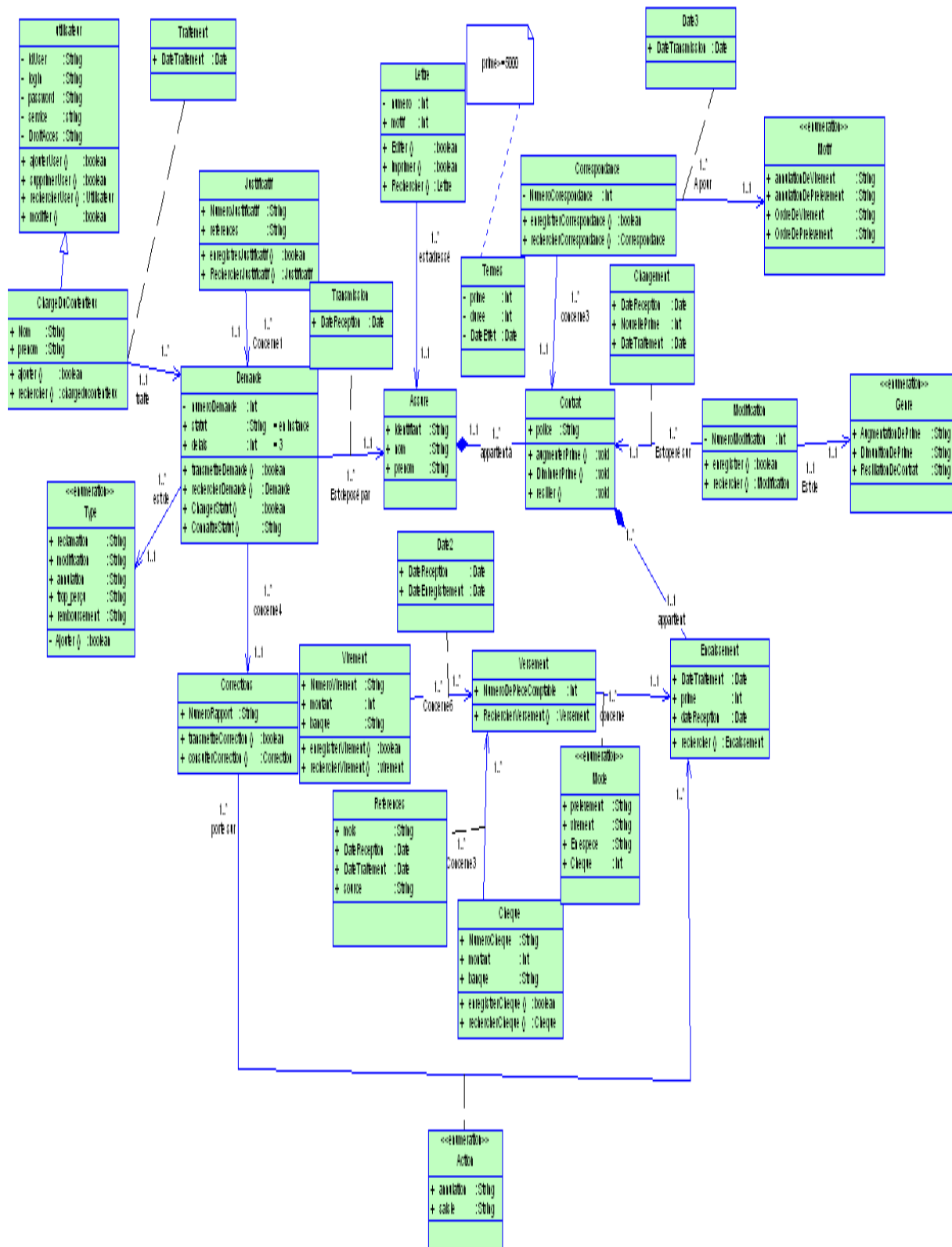


II.4 Diagramme de classes

1. Définition et objectifs

Le diagramme de classe permet de fournir une vue statique et abstraite d'un système. Il est le plus important de la modélisation orientée objet car il définit les objets du système qui vont interagir pour réaliser les cas d'utilisations.

2. représentation des classes



3. description des classes

Classe : Utilisateur , l'ensemble des utilisateurs de l'application					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Private	IdUser	1.1	String	Identifiant de l'utilisateur	Frozen
Private	Login	1.1	String	Le nom de connexion	changeable
Private	Password	1.1	String	Le mot de passe	changeable
Private	Service	1.1	String	Le service de l'utilisateur	Frozen
Private	droitAcces	1.*	String	Les droits d'accès d'un utilisateur	Frozen
Méthodes					
Visibilité	Nom	Type retour	Description		
Public	AjouterUser ()	boolean	Permet d'ajouter un utilisateur		
Public	SupprimerUser()	boolean	Permet de supprimer un utilisateur		
Public	RechercherUser()	Utilisateur	Permet de rechercher un utilisateur		
Public	ModifierUser ()	boolean	Permet de modifier un utilisateur		

Classe : ChargeDuContentieux , regroupe les personnes chargées de gérer le contentieux					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Public	Nom	1.1	String	Nom de celui qui traite la demande	Frozen
Public	Prénom	1.1	String	Prénom de celui qui traite la demande	Frozen
Méthodes					
Visibilité	Nom	Type retour	Description		
Public	Ajouter ()	boolean	Permet au chargé du contentieux d'associer son nom avant de traiter une demande		
Public	Rechercher ()	ChargeDuContentieux	Permet de rechercher le chargé du contentieux		

Classe : Demande, demande déposée par l'assuré					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Public	NumeroDemande	1.1	Int	Indique le numéro de la demande	Frozen
Public	Statut	1.1	String	Donne le niveau de traitement de la demande	Changeable
Public	Délais	1.1	Int	Donne le délai de traitement de la demande	Changeable
Public	DateReception	1.1	Date	Indique la date de réception de la demande	Frozen
Public	Type	1.1	String	Indique le type de la demande	Frozen
Public	DateTraitement	1.*	Date	Indique la date de traitement de la demande	Frozen

Méthodes			
Visibilité	Nom	Type retour	Description
Public	TransmettreDemande ()	Boolean	Permet de transmettre une demande
Public	RechercherDemande ()	Demande	Permet de rechercher une demande
Public	ChangerStatut ()	boolean	Permet de changer le statut de la demande
Public	ConnaitreStatut ()	String	Permet de connaître le niveau de traitement d'une demande

Classe : Justificatif, justificatif d'une demande					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Public	NumeroJustificatif	1.1	String	Indique le numéro de justificatif de la demande	Frozen
Public	Références	1.*	String	Donne les références d'un justificatif	Changeable
Méthodes					
Visibilité	Nom	Type retour	Description		
Public	EnregistrerJustificatif()	Boolean	Permet d'enregistrer un justificatif		
Public	RechercherJustificatif()	Justificatif	Permet de rechercher un justificatif		

Classe : Assuré , l'ensemble des assurés de l'UAB-vie					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Public	Identifiant	1.1	String	Identifiant de l'assuré	Frozen
Public	Nom	1.1	String	Nom de l'assuré	Frozen
Public	Prénom	1.*	String	Prénom de l'assuré	Frozen

Classe : Lettre , lettre à envoyer à l'assuré après traitement de sa demande					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Public	Numéro	1.1	Int	Numéro d'une lettre	Frozen
Public	Motif	1.1	Int	Motif d'une lettre	Frozen
Méthodes					
Visibilité	Nom	Type retour	Description		
Public	Editer ()	Boolean	Permet d'éditer une lettre		
Public	Imprimer ()	Boolean	Permet d'imprimer une lettre		
Public	Rechercher ()	Lettre	Permet de rechercher une lettre		

Classe : Correspondance , lettre à envoyer aux sources de paiements					
Attributs					
Visibilité	Nom	Multiplicité	Type	description	propriété
Public	numeroCorrespondance	1.1	String	Correspond au numéro de la correspondance	Frozen
Public	Motif	1.1	String	Texte expliquant le motif de la correspondance	Frozen
Public	DateTransmission	1.1	Date	Date de transmission de la correspondance	Frozen
Méthodes					
Visibilité	Nom	type retour		Description	
Public	enregistrerCorrespondance()	boolean		Enregistre une correspondance dans la base de données	
Public	rechercherCorrespondance()	Correspondance		Recherche une correspondance en fonction des paramètres	

Classe : contrat, contrats des assurés avec l'UAB-vie					
attributs					
visibilité	Nom	Multiplicité	Type	description	propriété
Public	police	1.1	String	Correspond au numéro de police du contrat	Frozen
Private	prime	1.1	int	Correspond à la prime mensuelle	Frozen
Private	duree	1.1	String	Durée totale du contrat	Frozen
Private	dateEffet	1.1	Date	Date à partir de laquelle l'assuré commence à cotiser	Frozen
méthodes					
Visibilité	Nom	Type retour	Description		
Public	rechercherContrat ()	Contrat	Permet de rechercher un contrat en fonction des paramètres		

Classe : Modification					
Attributs					
Visibilité	Nom	multiplicité	type	description	Propriété
Private	NumeroModification	1.1	Int	Correspond au numéro de modification d'un contrat	Frozen
Public	Genre	1.1	String	Correspond au type de modification	Frozen
Public	DateTransmission	1.1		Date de transmission de la correspondance	Frozen
Méthodes					
Visibilité	Nom	Type retour	Description		
Public	Enregistrer ()	boolean	Permet d'enregistrer une modification		
Public	Rechercher ()	Modification	Permet de rechercher une modification		

Classe : Encaissement , contient les différents encaissements d'un assuré					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriétés
Public	DateTraitement	1.1	Date	Indique la date de traitement d'un encaissement	Frozen
Private	Prime	1.*	Int	Indique la somme encaissée	Changeable
Public	DateReception	1.1	Date	Indique la date d'encaissement	Frozen
Public	mode	1.1	String	Le mode de paiement	Frozen
Méthodes					
Visibilité	Nom	Type retour	Description		
Public	Rechercher ()	Encaissement	Permet de rechercher un encaissement		

Classe : Chèque , chèques envoyés par les banques et par certains assurés					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriétés
Public	numeroCheque	1.1	String	Donne le numéro de chèque	Frozen
Private	montant	1.1	int	Donne le montant du chèque	Frozen
Private	banque	1.1	String	Donne la banque de prélèvement	Frozen

Public	DateReception	1.1	Date	Indique la date de réception du chèque	Frozen
Public	dateEnregistrement	1.1	Date	Indique la date d'enregistrement du chèque	Frozen
Public	Source	1.1	String	Indique le propriétaire du chèque	Frozen
méthodes					
Visibilité	Nom		Type retour	Description	
Public	enregistrerChèque ()		boolean	Permet d'enregistrer un chèque	
Public	rechercherChèque ()		Chèque	Permet de rechercher un chèque	

Classe : virement , virement fait par les banques sur les comptes des assurés					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriétés
Private	NumeroVirement	1.1	String	Numéro de virement bancaire	Frozen
Public	Montant	1.1	Int	Montant viré par la banque	Frozen
Public	Banque	1.1	String	La banque faisant le virement	Frozen

Public	DateReception	1.1	Date	Date de réception du chèque	Frozen
Méthodes					
Visibilité	Nom	Type retour	Description		
Private	EnregistrerVirement()	Boolean	permet d'enregistrer les virements		
Public	RechercherVirement()	Virement	Permet de rechercher les virements		

Classe : Corrections , corrections faites par le chargé du contentieux					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriété
Public	NumeroCorrection	1.1	String	Numéro d'une correction donnée	Frozen
Public	Action	1.2	String	Indique s'il s'agit d'une annulation et/ou d'une saisie	Changeable
Méthodes					
Visibilité	Nom	Type retour		Description	
Public	TransmettreCorrection()	Boolean		Permet de transmettre une correction	
Public	ConsulterCorrection()	Correction		Permet de consulter les corrections	

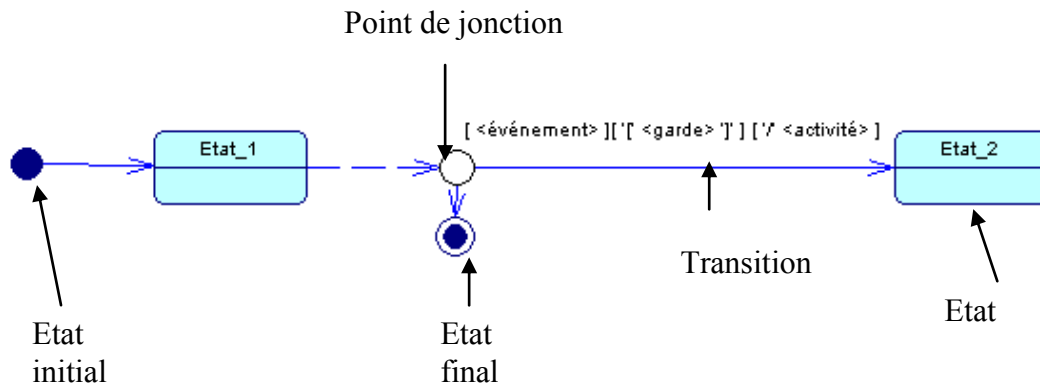
Classe : Versement , versement fait au compte de l'UAB-vie					
Attributs					
Visibilité	Nom	Multiplicité	Type	Description	Propriétés
Public	numeroDepieceComptable	1.1	Int	Numéro attribué à une pièce comptable	Frozen
Public	Mois	1.1	String	Mois concerné par le versement	Frozen
Public	DateReception	1.1	Date	Date à laquelle la pièce comptable a été reçue	Frozen
Public	DateTraitement	1.1	Date	Date	Frozen
Méthodes					
Visibilité	Nom	Type retour		Description	
Public	RechercherVersement()	Versement		Permet de rechercher un versement	

II.5 Diagramme d'état-transitions

Les diagrammes d'état-transition permettent de décrire le comportement interne d'un objet à l'aide d'automate à états finis. Il permet de décrire le cycle de vie d'un objet d'une manière séquentielle. A la différence du diagramme d'activité le diagramme d'état-transition ne permet pas de voir le comportement global du système mais d'un seul composant du système.

1. concepts et formalisme

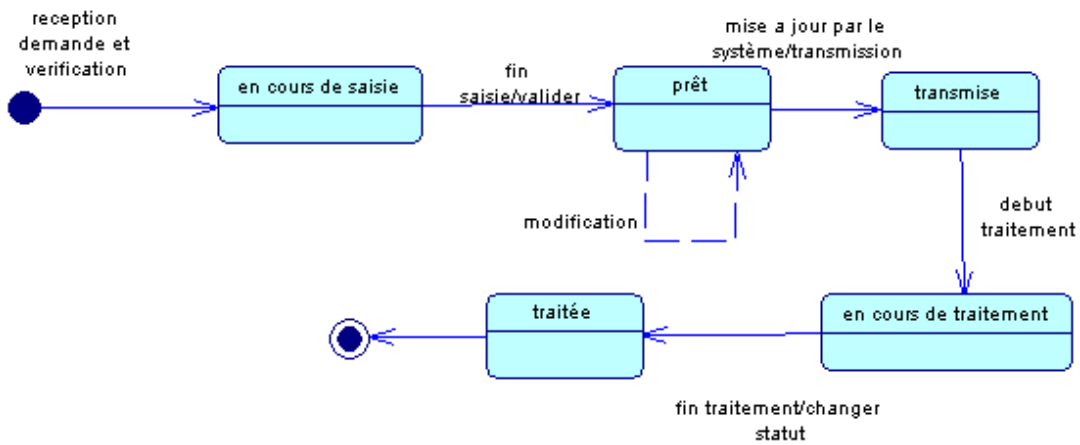
- **état** : un état représente une période de vie d'un l'objet pendant laquelle l'objet attend un évènement ou accomplit une activité.
- **état initial** : c'est le premier état d'un objet. L'état initial est u pseudo état de départ par défaut, lorsque le diagramme d'états-transitions est invoqué.
- **état final** : c'est un pseudo état qui indique la fin du diagramme d'états-transitions.
- **évènement** : un évènement est une action qui provoque le déclenchement d'une transition susceptible de faire passer l'objet d'un état à un autre. On distingue les types d'évènement suivants :
 - ✓ **signal** : véhicule une communication asynchrone a sens unique entre deux objets.
 - ✓ **appel** : représente la réception d'un appel d'une opération par un objet
 - ✓ **changement** : générer par un objet pour traduire sa satisfaction face à une expression booléenne.
 - ✓ **temporel** : générer par le passage du temps. La date peut être indiquée de manière absolue (date précise) ou de manière relative (temps écoulé).
- **transition** : définit la réponse d'un objet suite à un évènement.
- **point de choix** : les points de choix sont utiliser pour indiquer les différentes alternatives pour le franchissement d'une transition.
- **garde** : correspond à une condition d'exécution d'une transition
- **activité** : correspond à une opération exécutée dans un état



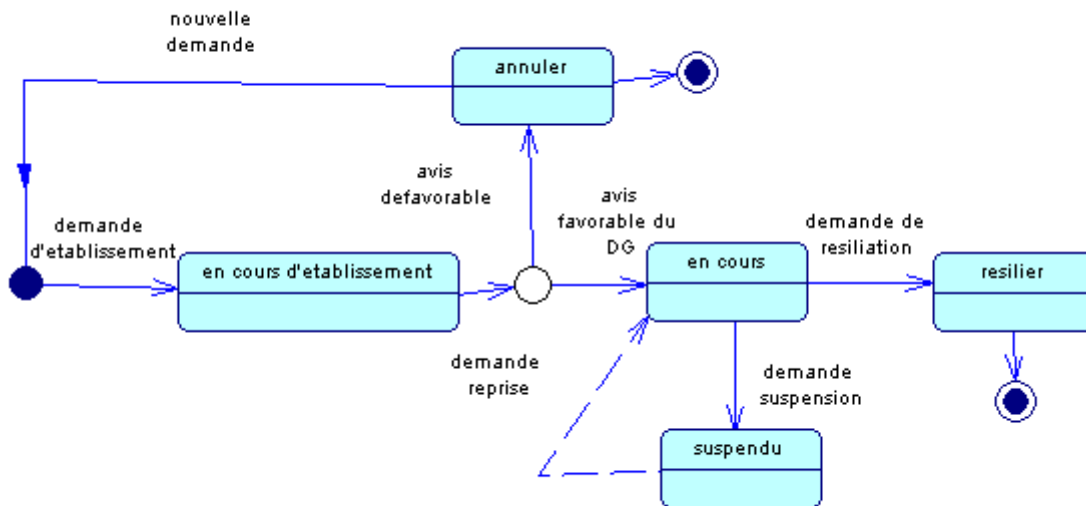
Représentation graphique d'un diagramme d'état transition

2. représentation des diagrammes d'état-transition

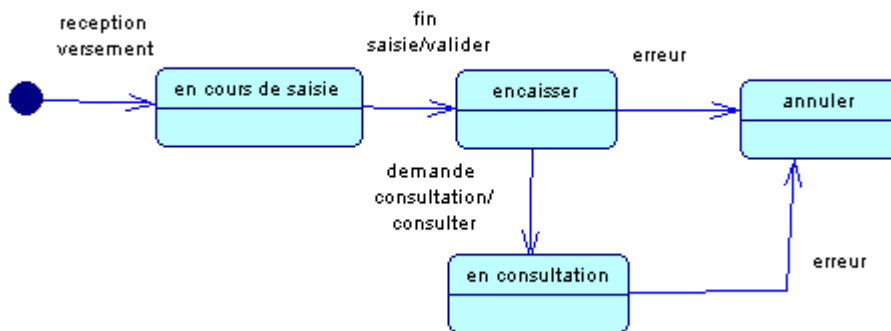
- classe : Demande



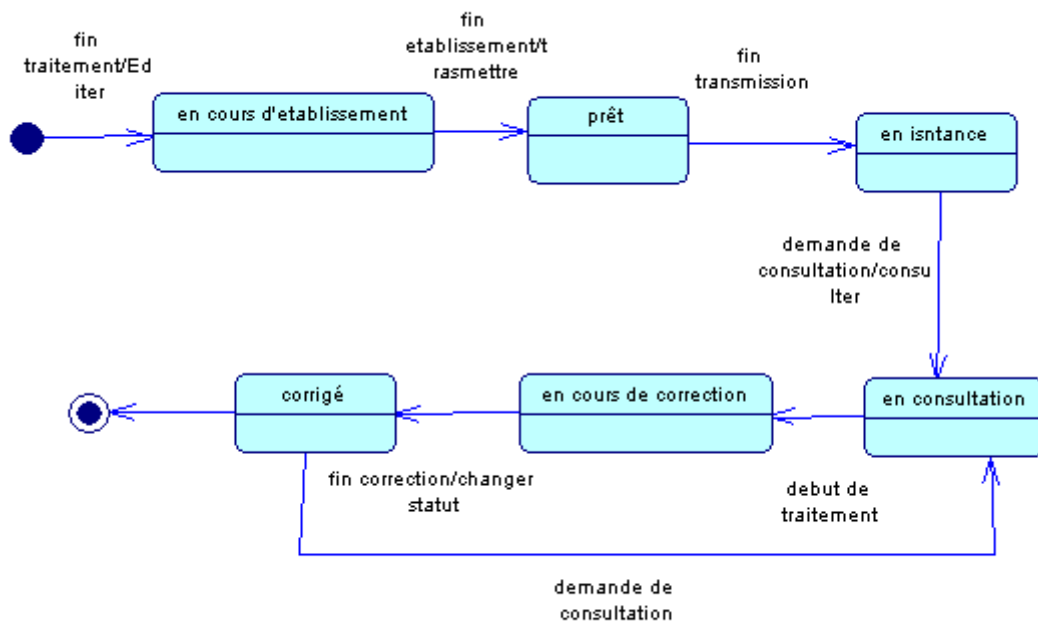
- classe : Contrat



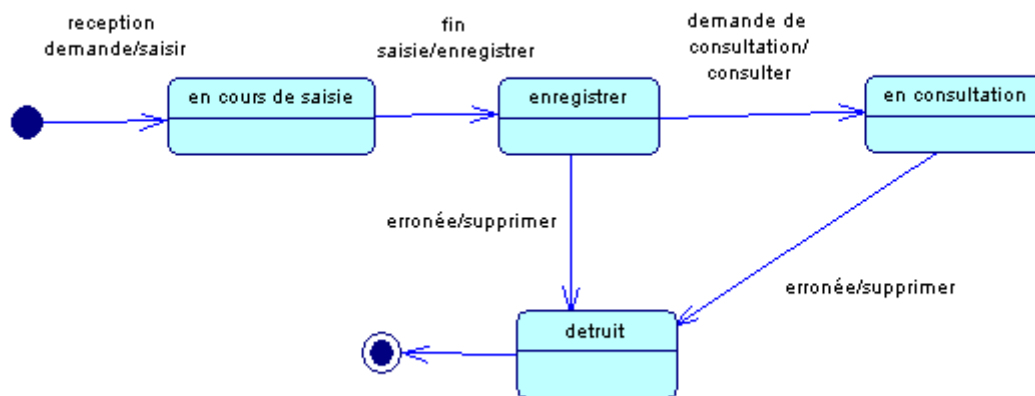
- **classe** : Encaissement



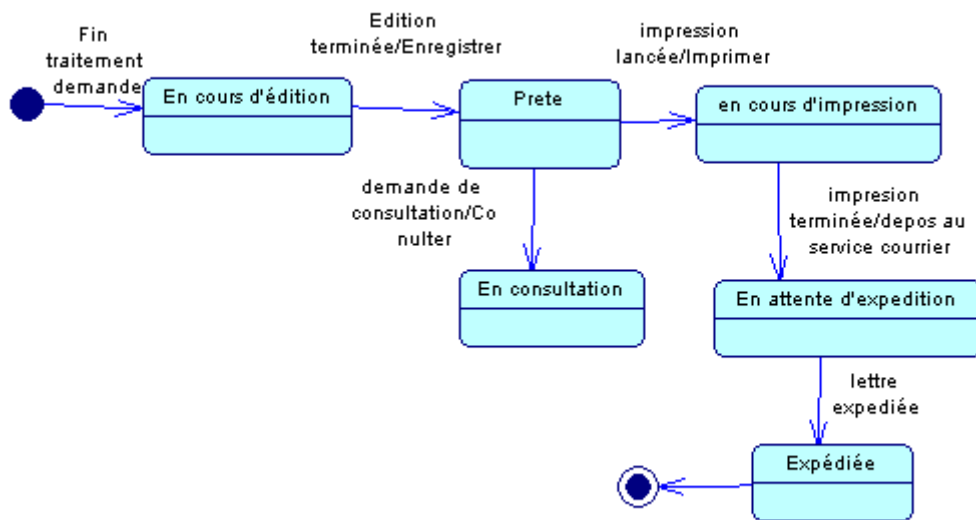
- **classe** : Corrections



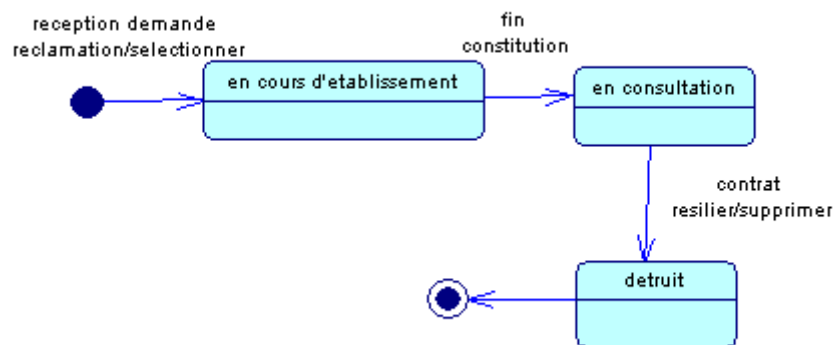
- classe : Correspondance



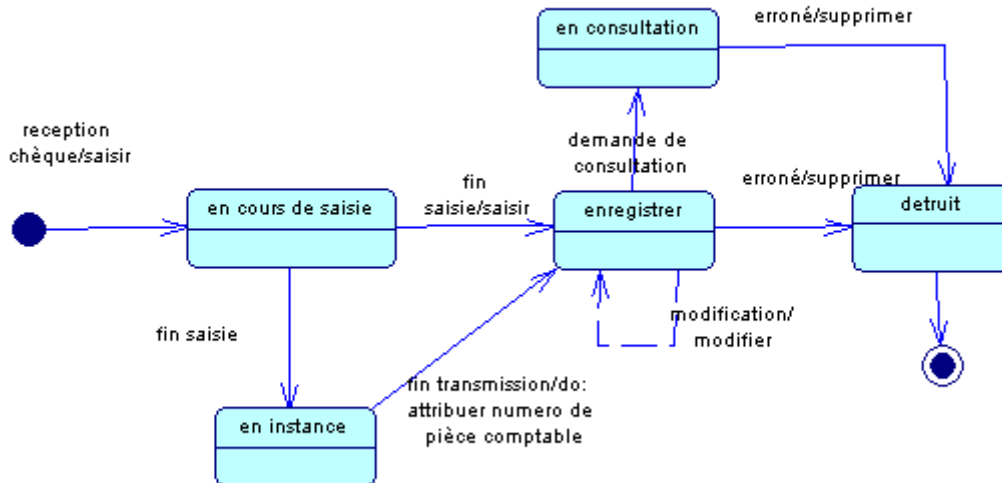
- classe : Lettres



- classe : Référentiel



- **classe** : Chèque



III. Etude technique

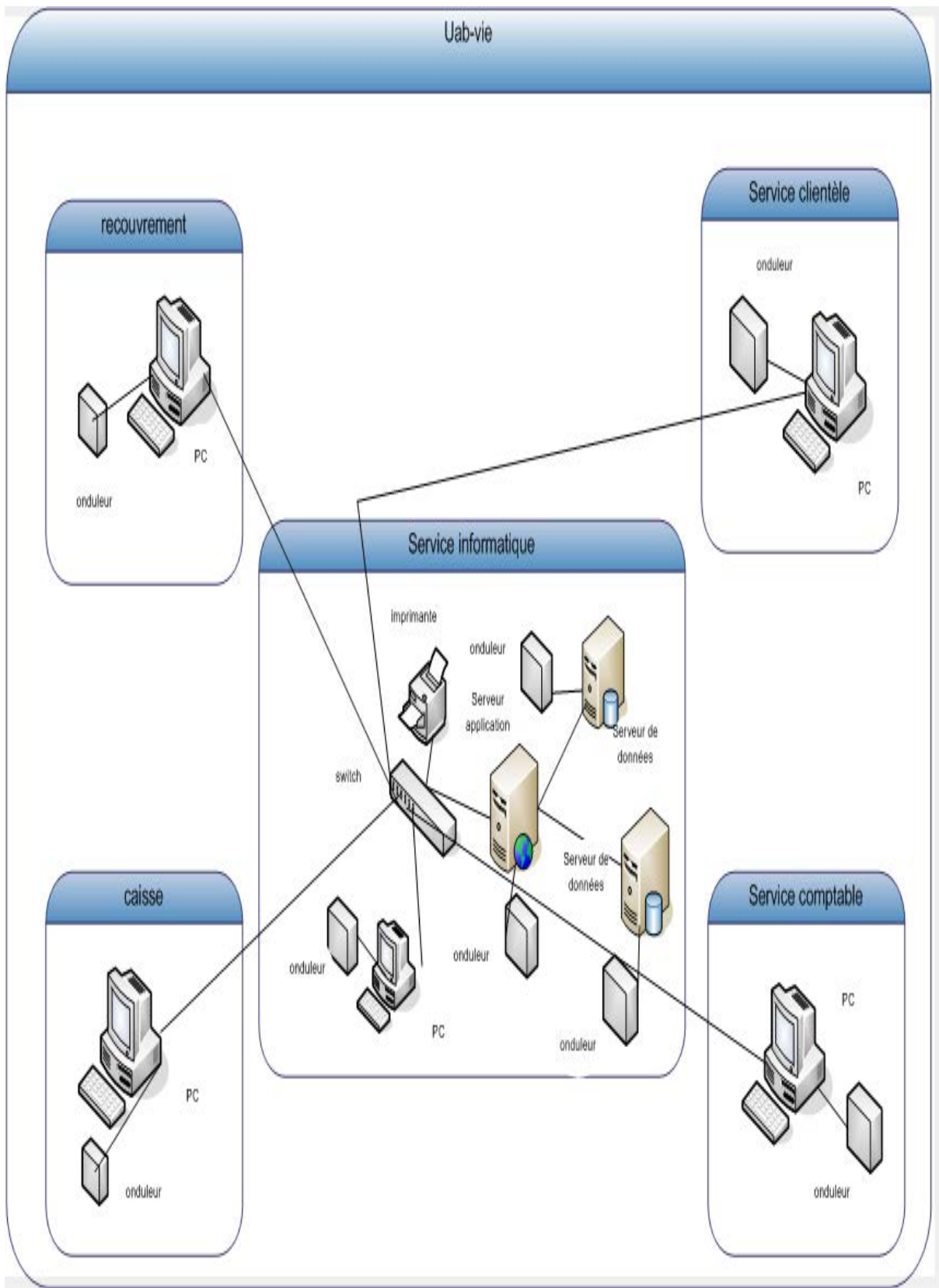
III.1 Description du scénario retenu

Ce scénario est identique aux scénarii à la différence que chaque service enregistre les documents qui lui parviennent. L'application est de type Client/Serveur et développée en php. Elle est accessible à travers le réseau local via un serveur web apache situé au service informatique.

Le pointage est automatisé et les corrections sont transmises à l'informatique pour la mise à jour. Le SVC pourra transmettre les différentes requêtes au RC et suivre l'évolution du traitement de ces requêtes.

Le comptable enregistre les entrées de chèques et les ordres de virements. La caissière est chargée de transmettre les cas de paiement par chèque au comptable pour l'attribution des numéros de pièces comptables. Ces différents enregistrements sont accessibles par le RC leur permettant ainsi de connaître les partenaires qui sont à jour. L'application permettra d'avoir un suivi des ordres de modification de prime, d'annulation de contrat et gérer les ordres de prélèvement pour les nouvelles productions. L'application accède également à deux bases de données : une base pour le pointage (existante) et une base pour le suivi des documents utilisés.

III.2 représentation du flux d'information



III.3 les coûts

1. coûts matériels

Désignation	Caractéristiques	Quantité	Prix unitaire	Prix total
Serveur de base de données	Processeur : 2,4 GHz, bus frontal 1066 MHz, 4 Mo de mémoire cache. Mémoire de masse : 587 Go	1	1. 1600000	2. 1600000
Micro-ordinateur	Pentium IV	05	Existant	
Onduleur	Autonomie d'au moins une heure	08	144534	1156272
Switch	24 ports	01	149231	149231
imprimante	HP laser 4200	01	Existant	
Serveur web	Apache	01	Gratuit	
Système d'exploitation	Windows XP/Linux	01	Existant	
Anti-virus	Avast	01	Existant	
Environnement de développement	Macromedia	01	Gratuit	
Sgbd	MySQL	01	Existant	
Total				2905503

2. coûts de développement

La méthode utilisée pour le calcul du coût de développement est COCOMO. C'est une méthode qui permet de calculer l'effort à consentir pour le développement d'un logiciel ainsi que le temps nécessaire en fonction de plusieurs paramètres. Il s'agit notamment de la complexité de l'application et du nombre de ligne de code que l'application devrait contenir.

Les formules utilisées pour notre cas espèce sont les suivantes :

$$S: \text{Effort} = 2,4 * \text{KLS}^{1,05} \quad T_{\text{Dev}} = 2,5 * \text{Effort}^{0,38}$$

$$P: \text{Effort} = 3 * \text{KLS}^{1,12} \quad T_{\text{Dev}} = 2,5 * \text{Effort}^{0,35}$$

$$E: \text{Effort} = 3,6 * \text{KLS}^{1,2} \quad T_{\text{Dev}} = 2,5 * \text{Effort}^{0,32}$$

Unité :

Effort en **HommeMois**.

TDev en **Mois**

KLS= Kilo Ligne Source représente le nombre de milliers d'instructions que contiendra l'application.

Application :

L'application à développer pour l'automatisation du contentieux est de complexité S vu qu'il y'a beaucoup de contraintes et de cas à gérer.

Pour ce qui concerne le nombre de lignes du code, nous l'estimons à 5100 KLS .Ce qui donne le calcul suivant :

Effort à consentir :

$$\text{Effort} = 2,4 * (5100 / 1000)^{1,05} = 13,27 \text{ HM}$$

Temps de développement requis

$$T_{\text{Dev}} = 2,5 * (13,27)^{0,38} = 6,67 \text{ mois}$$

Nombre de personne nécessaire pour le développement

$$\text{NBpersonne} = \text{Effort} / T_{\text{Dev}} = 13,27 / 6,67 = 1,98 \sim 2 \text{ personnes}$$

Coût de l'application

$$\text{Coût de développement} = 6,67 * 200000 * 2 = 2668000 \text{ FCFA}$$

3. coûts de formation

Coût par personne	Nombre d'utilisateurs	Coût total
5000	8	40000 FCFA

4. coût total du projet

Désignation	Coût (FCFA)
Matériels	2905503
Développement	2668000
Formation	40000
Total	5613503

III.2 Langages et Outils de développement

1. Serveur web Apache

Apache est un serveur web conforme au protocole http. Le nom Apache vient du mot "a patchy server", en raison du mode de développement par patches de ces premières versions.

Apache existe sur plusieurs plates-formes (UNIX et Windows pour le moment) et permet d'exécuter des applications développées en PHP.

Ses performances, sa robustesse, son utilisation aisée, et sa licence en font le serveur web le plus populaire au monde depuis mars 1997 (actuellement **53.44%** des sites mondiaux tournent sur Apache, soit **1.182.142** sites répertoriés).

Apache intègre le concept de modules, ce qui rend facile le développement d'extensions par la communauté du logiciel libre. Parmi tous les modules d'Apache, on peut citer:

- le module **typo**: il permet de corriger à la volée une URL mal formée, et de rediriger la requête vers une URL valide approchante.
- le module **include**: il permet d'insérer des directives (**server side includes**) dans les documents HTML, et de demander au serveur d'interpréter ces dernières. Les directives incluent des tests et permettent principalement d'inclure des fichiers dans des pages HTML.
- le module **log**: il permet de gérer les logs (fichiers dans lesquels tous les accès au serveur sont archivés), et d'ajouter dans ces derniers des informations propres à l'utilisateur.
- le module **auth**: ce module permet une authentification du client grâce au protocole HTTP. Cette authentification n'est pas forcément très fiable puisque le

protocole envoie le mot de passe **en clair**. L'utilité est d'isoler du reste du serveur des pages utiles à un faible nombre de personnes.

- le module **status**: ce module permet de générer une page HTML contenant le statut du serveur. Des informations telles que le nombre de serveurs lancés ou les dernières requêtes reçues sont disponibles
- le module **info**: ce module qui ressemble au précédent permet de générer une page HTML contenant des informations sur la configuration du serveur.
- le module **so**: ce module expérimental permet de charger "à la volée" sans recompilation des modules, en utilisant le principe des bibliothèques dynamiques

2. architecture 3-tier

L'architecture 3-tier ou architecture à trois niveaux est une architecture de type client /serveur. C'est un système dans lequel l'utilisateur de l'application appelé client demande une ressource à un serveur d'application appelé middleware. Le serveur d'application fournit la ressource mais en faisant appel à un serveur secondaire généralement un serveur de base de données. Donc l'architecture 3-tier distingue 3 niveaux distincts au sein d'une application ou d'un système. Chaque niveau ayant un rôle bien précis.

L'architecture 3-tier présente plusieurs avantages pouvant se résumer en trois (03) points :

- La souplesse et la flexibilité
- Une sécurité accrue car elle est définie par niveau.
- Une grande performance car les tâches sont partagées.

IV. Protection de l'application

Le système à mettre en place présente de nombreux avantages, mais reste tout de même exposé à de nombreux risques. Les plus fréquents sont :

- Panne matériel du système
- Infection du système par un virus
- Panne réseau ou électricité
- Attaques

IV.1 Procédures de secours

1. Panne réseau ou électricité

- **Electricité** : l'UAB ne dispose pas de groupe de relais en cas de panne d'électricité, nous préconisons l'utilisation d'onduleurs.

En cas de panne d'électricité, les utilisateurs auront le temps pour sauvegarder leurs données et effectuer les opérations urgentes.

- **Réseau** : une panne du Switch du réseau local rend indisponible le système. Dans ce cas, nous préconisons que le travail s'effectue manuellement jusqu'au rétablissement du réseau.

2. Panne matériel ou système

- **Poste de travail**: le matériel de travail peut souvent tomber en panne. En cas de panne d'un poste de travail, il faut tout simplement utiliser un autre en attendant la réparation du poste en panne ou son remplacement
- **En cas de panne d'un serveur web**, nous ne préconisons que le travail s'effectue manuellement en attendant la réparation ou le remplacement du dit serveur.
- **En cas de panne du serveur de données**, nous préconisons d'utiliser le serveur relais .En effet l'UAB-vie dispose de deux serveurs de données pour prévenir les pannes de serveurs.
- **Système** : Une panne du système bloque totalement le traitement automatique du contentieux. En cas de plantage de l'application, nous suggérons sa réinitialisation. Si le problème persiste, nous suggérons de faire appel à l'informaticien ou aux développeurs.
En cas d'indisponibilité générale, il faudrait tout simplement faire recours au traitement manuel.

IV.2 Procédures de sécurité

1. Protection contre les virus

Les virus sont des petits programmes susceptibles de nuire au bon fonctionnement d'un système. Bien que l'application ne soit pas ouverte sur Internet, il convient de prendre des mesures contre les virus car les agents utilisent de plus en plus de clés usb.

Pour cela nous suggérons de payer une nouvelle licence de l'anti-virus Avast et de faire la mise à jour régulièrement.

2. Politique de sauvegarde

Etant donné que l'UAB-vie gère des données très sensibles susceptibles de créer des conflits, il est important d'avoir une politique des données très fiables et efficaces. Pour cela nous suggérons que :

- les agents continuent les enregistrements manuels des paiements ;
- effectuer chaque jour des sauvegardes de la base de données sur le serveur de relais ;

- faire régulièrement la sauvegarde de la base de données sur un disque externe ou sur des CD.

3. Protection contre les attaques malveillantes

Bien que l'application ne soit pas ouverte sur Internet, il est nécessaire de prendre des mesures contre les attaques malveillantes.

En effet, une attaque peut être interne et souvent plus dangereuse que les attaques à distance car les attaquants internes disposent de plus d'opportunité et d'informations sur le système. Nous avons opté pour les mesures suivantes :

- **L'obligation de connexion**

Tout utilisateur désirant effectuer une quelconque opération doit s'authentifier à l'aide d'un login et d'un mot de passe. Cela permettra de suivre le profil de chaque utilisateur.

- **Le contrôle des données saisies lors des connexions.**

Les utilisateurs pouvant profiter de la richesse du langage SQL pour injecter des codes malveillants des mesures prises pour contrôler la longueur des mots de passe et des logins.

- **Cryptage des mots de passe**

Si l'attaquant a accès à la base de données, il peut vouloir récupérer les logins et les mots de passe dans la base de données. Une solution est le cryptage des logins et mot de passe à l'aide de Visual Studio.

- **La protection de la base de données**

Tout accès à la base de données passe par une authentification

IV.3 Procédures de mise en œuvre

1. tests

Comme le préconise la démarche que nous utilisons, l'application doit être testée pendant au moins trois (3) mois avant sa mise en œuvre définitive. Les utilisateurs devront faire appel aux développeurs, ils détectent des failles durant la période de test.

2. formation des utilisateurs

L'application agit beaucoup avec l'utilisateur ; il convient alors de former les utilisateurs afin qu'ils puissent bien effectuer les tests.

Conclusion

L'étude menée durant ce stage a révélée que le contentieux souffre actuellement d'une gestion manuelle et apporte une solution adaptée à la politique futuriste de l'UAB-vie. L'étude préconise l'entrée des différentes réclamations au niveau du service clientèle qui les transmet au recouvrement et contentieux à travers la future application. Le recouvrement aura pour tache de traiter ces réclamations et de les transmettre, à travers la future la future application, au service informatique pour correction.

L'étude a permis d'élaborer le cahier des charges qui servira de référence pour la construction du futur système. Elle a permis notamment de construire des diagrammes UML présentant les différentes facettes de la structure du futur système, de prendre des mesures pour sa protection et de planifier les prochaines étapes qui aboutiront à la construction du futur système.

Ce stage nous a été un véritable cadre d'expression et nous a permis de mener un projet de bout en bout tout en respectant les spécifications de la méthode RUP et du langage de modélisation UML. Il a par ailleurs été mené dans un climat favorable avec la l'aide du groupe de pilotage et la complicité du personnel de l'UAB-vie.

Annexes

Ecran des maquettes :

1. écran de connexion



The screenshot shows a login interface with a green header bar. The text 'UAB-vie' is displayed in large green letters, with the tagline 'la certitude d'une vie sereine' below it. To the right, the text 'Gestion du Contentieux 1.0' is visible. The login fields are labeled 'Login' and 'Password', each followed by a grey input box. A 'connexion' button is centered below the fields. At the bottom, the authors are listed as 'auteurs: Sinare Judith et Tapsoba Abdou-malick ESI-AP3'.

UAB-vie
la certitude d'une vie sereine

Gestion du Contentieux 1.0

Login

Password

connexion

auteurs: Sinare Judith et Tapsoba Abdou-malick ESI-AP3

2. écran d'accueil recouvrement



3. écran d'accueil service clientèle



4. écran de transmission d'une demande

UAR-vie
la certitude d'une vie sereine

Gestion du Contentieux 1.0

menu

- Dossiers
- reclamation
- annulation
- modification
- rechercher
- Transferts
- deconnexion

Type*

ID Client*

Police*

Prime*

Societe*

Mode Paiement*

Echéances/periode mois année

les champs avec * sont obligatoires

5. écran de d'enregistrement d'un cheque

The screenshot shows a web interface for 'UAB-vie' with the tagline 'la certitude d'une vie sereine'. The page title is 'Gestion du Contentieux 1.0'. On the left is a navigation menu with options: Dossiers, cheque (highlighted), correspondance, Transferts, Partenaires, rechercher, and deconnexion. The main form area contains the following fields:

- Nunero* : text input field
- Montant* : text input field
- Beneficiaire* : dropdown menu with 'ESI' selected
- Banque* : dropdown menu with 'BOA' selected
- Mois Cotisation* : dropdown menu with 'janvier' selected
- Date Reception* : date selection with 'jour' 1, 'mois' janvier, and 'année' 2007
- Date Enregistrement* : date selection with 'jour' 1, 'mois' janvier, and 'année' 2007

An 'ENREGISTRER' button is located at the bottom right of the form.

6. écran du résultat de la recherche des partenaires non a jour pour le mois de janvier

menu

Dossiers

cheque
correspondance

Transferts

Partenaires

rechercher

deconnexion

JANVIER

SOCIETE:

ADRESSE

TELEPHONE

sonabel

11 avnue boumerdes

50426872

onea

98 avenue liberte

50498725

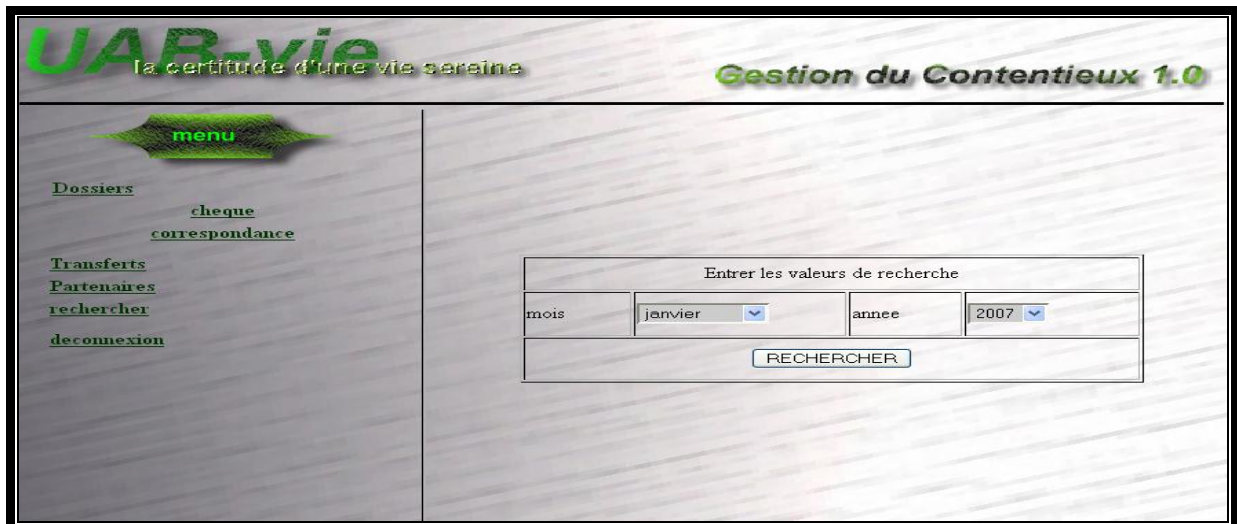
bacb

45 avenue liberte

6547243

IMPRIMER

7. écran de recherche



Bibliographie

<http://www.bd.enst.fr>
<http://www.commentcamarche.net>
<http://www.developpez.com>
<http://www.alaide.com/cours.php>

De MERISE à UML de Pascal PARE
UML 2.0

Projet de fin de cycle Da Santoro Patrick
Projet de fin de cycle Ouattara Yacouba
Projet de fin de cycle Konkole Evariste