

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



Université Mouloud Mammeri
Faculté de Génie Electrique et de l'Informatique
Département d'Informatique
Tizi-Ouzou

Mémoire de fin d'études

En vue de l'obtention d'un diplôme de Master en
Informatique
Option : « Système Informatique »

Thème :

**Conception et réalisation d'une application client/server
multi-dossier pour le suivi de la comptabilisation des
stocks des produits finis**

Cas : ENIEM Tizi-Ouzou

Présenté par :

M^{elle} : BEN CHIKH Aicha

M^{elle} : HIRECHE Chabha

Dirigé par :

M^r HAMITI M.

--- Année universitaire : 2012-2013---

SOMMAIRE

Introduction générale

Chapitre I : Partie 1 : Présentation de l'organisme d'accueil.

I.	Présentation de l'organisme d'accueil.....	01
I.1	Historique.....	01
I.2	Mission et objectifs de l'ENIEM.....	01
I.3	Organigramme de l'organisme d'accueil.....	02
II.	Organigramme de l'unité commerciale.....	04
III.	Organigramme de département finance et comptabilité.....	06

Chapitre I : Partie 2 : Généralités sur les stocks.

I.	Généralité sur les stocks.....	07
I.1	Définition d'un stock.....	07
I.2	Définition de la gestion des stocks.....	07
II.	Nécessité d'une bonne gestion des stocks.....	09
III.	La comptabilité des stocks.....	09
III.1	Aspects comptable de la gestion des stocks.....	10
III.1.1	L'inventaire permanent.....	10
III.1.2	La valorisation des entrées/sorties.....	10
III.1.3	Les méthodes d'épuisement des stocks.....	11
III.1.4	Les méthodes du coût unitaire moyen pondéré (CUMP).....	11
III.1.5	L'écriture comptable.....	12
Conclusion.....		12

Chapitre II : Etat de l'art

Introduction	13
I. Description des procédures de travail.....	13
II. Positionnement du problème.....	13
III. Objectif de l'étude.....	13
Conclusion	14

Chapitre III : Analyse et spécification des besoins

Introduction	15
I. Identification des acteurs	15
I.1 Définition des acteurs	15
I.2 Formalisme graphique	15
I.3 Identification des acteurs.....	15
II. Identification des cas d'utilisation.....	16

II.1 Définition.....	16
II.2 Formalisme graphique.....	16
III. Les cas d'utilisation	17
III.1 diagramme des cas d'utilisation	18
III.2 Description des cas d'utilisation avec des scénarios	21
IV. Diagrammes de séquences et d'activités.....	23
Conclusion.....	31

Chapitre IV : Conception

Introduction	32
1. Architecture de l'application.....	32
2. Diagramme des classes.....	33
3. Conception de la base de données.....	34
a. Tableau récapitulatif des classes de la base de données	34
b. Le modèle logique de donnés.....	35
Conclusion	37

Chapitre V : Réalisation

Introduction	38
I. Description du matérielles utilisés.....	38
II. Présentation du langage de programmation utilisé.....	38
1. Notion de programmation visuelle et événementielle.....	38
2. Interface de développement « Delphi 7 ».....	39
3. Delphi et les bases de données.....	41
4. SQL Server 2005.....	41
III. Interfaces logiciel.....	43
Conclusion	48

Conclusion générale

Bibliographie

Webographie

Annexe A (UML)

Annexe B (les architectures logicielles)

LISTE DES FIGURES

Figure I.1 : Organigramme d'organisme d'accueil	02
Figure I.2 : Organigramme de l'unité commerciale	04
Figure I.3 : Organigramme du département finance & comptabilité	06
Figure III.1 : La démarche adoptée pour la modélisation	18
Figure III.2 : Diagramme de cas d'utilisation pour le comptable général	19
Figure III.3 : Diagramme de cas d'utilisation pour le cadre comptable	20
Figure III.4 : Diagramme de cas d'utilisation pour le chef de service comptabilité	20
Figure III.5 : Description du cas d'utilisation « S'authentifier ».....	21
Figure III.6 : Description du cas d'utilisation « Ajouter une entrée ».....	21
Figure III.7 : Description du cas d'utilisation «supprimer une sortie».....	22
Figure III.8 : Description du cas d'utilisation «Ajouter un article».....	22
Figure III.9 : Diagramme de séquence du cas d'utilisation « S'authentifier ».....	23
Figure III.10 : Diagramme d'activité du cas d'utilisation « S'authentifier ».....	24
Figure III.11 : Diagramme de séquence du cas d'utilisation « Ajout d'une entrée ».....	25
Figure III.12 : Diagramme d'activité du cas d'utilisation « Ajout d'une entrée ».....	26
Figure III.13 : Diagramme de séquence du cas d'utilisation « Ajout d'une sortie ».....	27
Figure III.14 : Diagramme d'activité du cas d'utilisation « Ajout d'une sortie »	28
Figure III.15 : Diagramme de séquence du cas d'utilisation « Consultation des états ».....	29
Figure III.16 : Diagramme d'activité du cas d'utilisation « Consultation des états ».....	29
Figure III.17 : Diagramme de séquence du cas d'utilisation « Impression d'un état ».....	30
Figure III.18 : Diagramme d'activité du cas d'utilisation « Impression d'un état ».....	31
Figure IV.1 : Architecture de l'application	32
Figure IV.2 : Diagramme de classe.....	33
Figure V.1 : Interface de développement « Delphi7 ».....	39

Figure V.2 : Interface d'authentification.....	43
Figure V.3 : Interface d'accueil.....	43
Figure V.4 : Interface d'ajout d'un article	44
Figure V.5 : Interface d'ajout d'une famille.....	44
Figure V.6 : Interface d'ajout d'un fournisseur.....	45
Figure V.7 : Interface d'ajout d'un dépôt.....	45
Figure V.8 : Interface d'ajout d'un client.....	46
Figure V.9 : Interface d'ajout d'une entrée.....	46
Figure V.10: Interface d'ajout d'une sortie.....	47
Figure V.11: Interface de recherche d'un mouvement de sortie.....	47
Figure V.12: Interface de journal des entrées.....	48

I INTRODUCTION GÉNÉRALE:

Dans tous les domaines notamment la gestion, l'être humain s'est mobilisé afin d'exploiter le temps comme il se doit.

Le bon fonctionnement d'une entreprise se base essentiellement sur une bonne organisation de ses services, éventuellement la gestion des stocks qui est l'élément crucial.

Dans notre cas, étant affectés à l'Entreprise Nationale des Industries en Electroménager (ENIEM), plus précisément, au sein de l'unité commerciale, le bon fonctionnement des mouvements des produits est d'une importance capitale dans la planification de la production ainsi que sur les ventes.

Notre projet consiste en la mise en place d'une application pour la comptabilisation des stocks des produits finis de l'ENIEM et porte sur l'implémentation d'une base de données et de développement d'une application client/serveur qui procurera un environnement convivial aux acteurs commerciaux et qui répondra aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Le présent rapport synthétise tout le travail que nous avons effectué dans cette perspective, il est organisé en cinq chapitres comme suit :

- § Le premier chapitre donne une présentation générale du projet : l'organisme d'accueil et les généralités sur les stocks.
- § Dans le second chapitre, nous procédons à un exposé de l'état de l'art du domaine qui nous concerne.
 - Nous présentons dans un premier temps le système existant pour dévoiler ses défaillances et ses limites.
 - Nous présentons également la solution que nous proposons afin de palier aux limites du système actuel.
- § Dans le troisième chapitre intitulé "Analyse et Spécification des besoins", nous présentons la démarche adoptée pour la modélisation de notre application en utilisant le langage de modélisation UML.
- § Dans le quatrième chapitre nous présentons la conception générale et la conception détaillée du système.
- § Le dernier chapitre décrit les tâches accomplies dans la réalisation.

Enfin nous donnons une conclusion récapitulant le travail réalisé sans oublier de citer les références bibliographiques sur lesquelles nous nous sommes appuyées pour mener à bien ce travail.

I. Présentation de l'organisme d'accueil : [B01]**I.1. Historique :**

L'Entreprise Nationale des Industries de l'Electroménager (ENIEM) est une entreprise de droit algérien constitué le 2 Janvier 1983, après la restructuration de SONELEC (Société Nationale de Fabrication et de Montage Matérielle Electrique et Electronique) suite au décret présidentiel N° 83 Janvier 1983.

De ce fait, l'ENIEM est passée d'une entreprise publique économique à une société par action en 1989, avec un capital social de 40.000.000 DA.

Dans ce cadre, l'ENIEM fut dotée de tous les organes de gestion légaux, à savoir :

- Ø Une assemblée générale.
- Ø Un conseil d'administration.
- Ø Un capital social.

I.2. Mission et objectifs de l'ENIEM :

La mission de l'ENIEM est d'assurer la production, la commercialisation, le développement et la recherche dans le domaine des différentes branches de l'électroménager. Comme, elle assure également la production des appareils de réfrigération, cuisson, climatisation et de lavage.

Parmi les principaux objectifs de l'entreprise, on trouve :

- Ø Amélioration de la qualité des produits.
- Ø Augmentation des capacités d'études et de développement.
- Ø Maitrise des couts de production.
- Ø Orientation de volume de production en corrélation avec les variations de la demande.
- Ø Placer son produit à l'échelle internationale.
- Ø Epargner au pays la sortie des devises (importation).

I.3. Organigramme de l'organisme d'accueil :

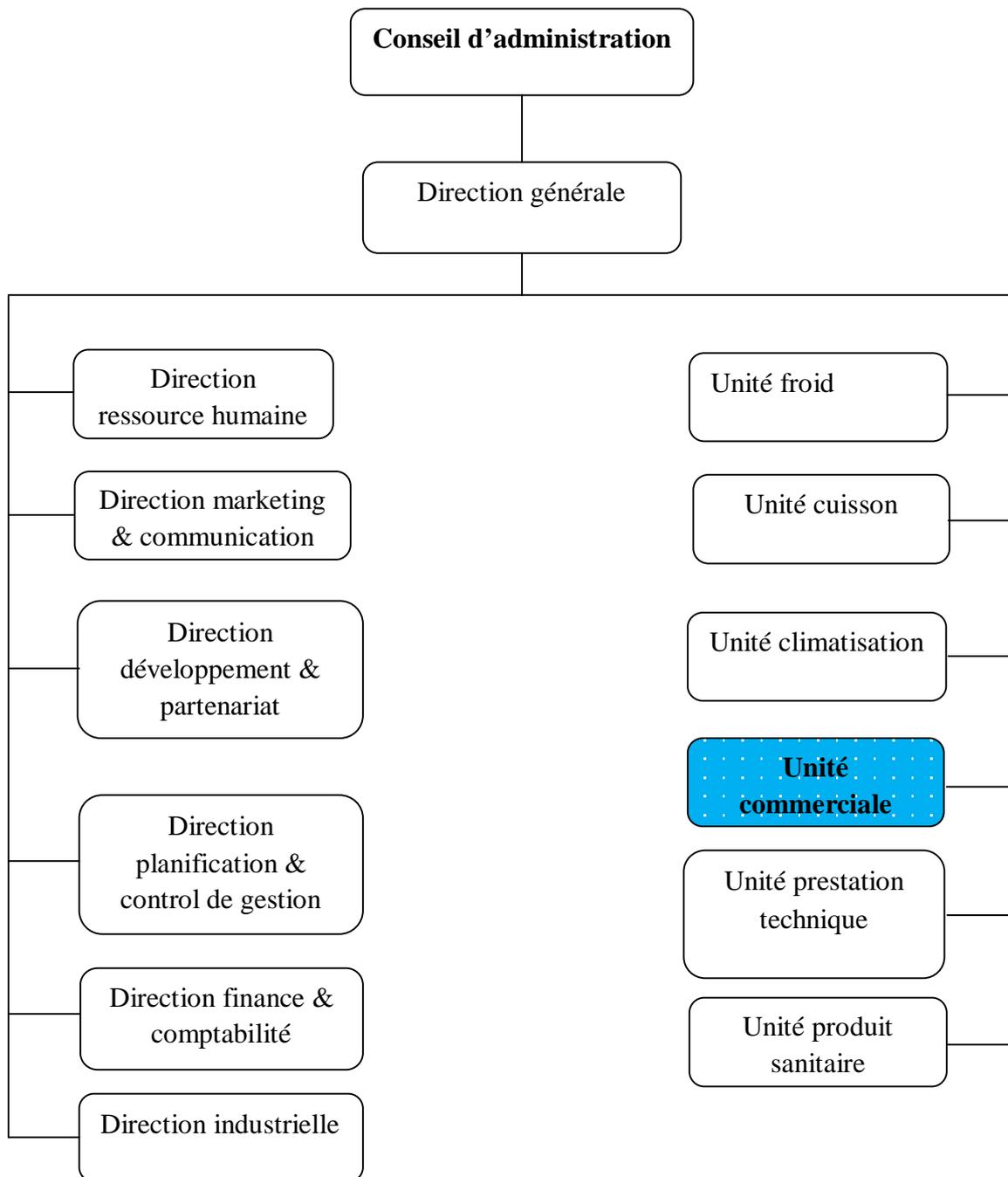


Figure I.1 : organigramme organisme d'accueil.

L'ENIEM est composée d'une direction générale et de six directions centrales définies comme suit :

- Ø **La direction générale** : elle est responsable de la stratégie et du développement de l'entreprise. Aussi, elle exerce son autorité hiérarchique sur l'ensemble des directions centrales et des unités.
- Ø **La direction des ressources humaines** : elle accroît la mobilisation et la valorisation du personnel dans ses actions au service du client. Elle s'occupe aussi du recrutement, l'accueil et l'information.
- Ø **La direction marketing & communication** : elle est chargée de la mise en œuvre de la stratégie concurrentielle de l'entreprise et de l'organisation des foires et salons nationaux.
- Ø **La direction développement et partenariat** : responsable des études et du développement du produit finis ainsi que du partenariat et de sous-traitance.
- Ø **La direction planification & contrôle de gestion** : responsable du contrôle de gestion, de l'audit finance et utilise les résultats de son dernier pour faire les recommandations nécessaires. Elle s'occupe aussi du budget de l'entreprise.
- Ø **La direction finance & comptabilité** : Composée de deux services, un service comptabilité générale de toute l'opération permettant l'établissement des comptes ; cette direction assure la vérification de l'application par la mise en œuvre d'un contrôle interne (règles comptable).
- Ø **La direction industrielle** : Elle est chargée de développer et de mettre en place les moyens et l'organisation industrielle nécessaire à la réalisation de la production en agissant sur l'approvisionnement ; les moyens et les techniques de production.

Aussi, l'entreprise est organisée en unités selon l'activité exercée. On trouve les unités suivantes :

- Ø **Unité froid** : elle est composée de trois lignes de production de plusieurs types de réfrigérateurs et congélateur :
 - Une ligne de réfrigérateurs petits modèles.
 - Une ligne de réfrigérateurs grands modèles.
 - Une ligne de congélateur.
- Ø **Unité cuisson** : elle est chargée de la production et du développement de produits de cuisson gaz, électrique ou mixte et tout produit de technologie similaire.
- Ø **Unité climatisation** : elle est responsable de la production et du développement de produits de climatisation et de chauffage.
- Ø **Unité commerciale** : assure les activités suivantes
 - La distribution et exportation des produits ENIEM.
 - Service après-vente.
- Ø **Unité prestation technique** : cette unité assure les fonctions de soutien aux unités de production dans les domaines de :
 - Réparation des outils.
 - Fabrication de pièces de rechanges mécaniques.
 - Conception et réalisation d'outillage.
 - Gestion des énergies et fluides.
 - Gardiennage et sécurité.

- Travaux d'imprimerie.
- Travaux de menuiserie.
- Travaux de nettoyage.

II. Organigramme de l'unité commerciale :

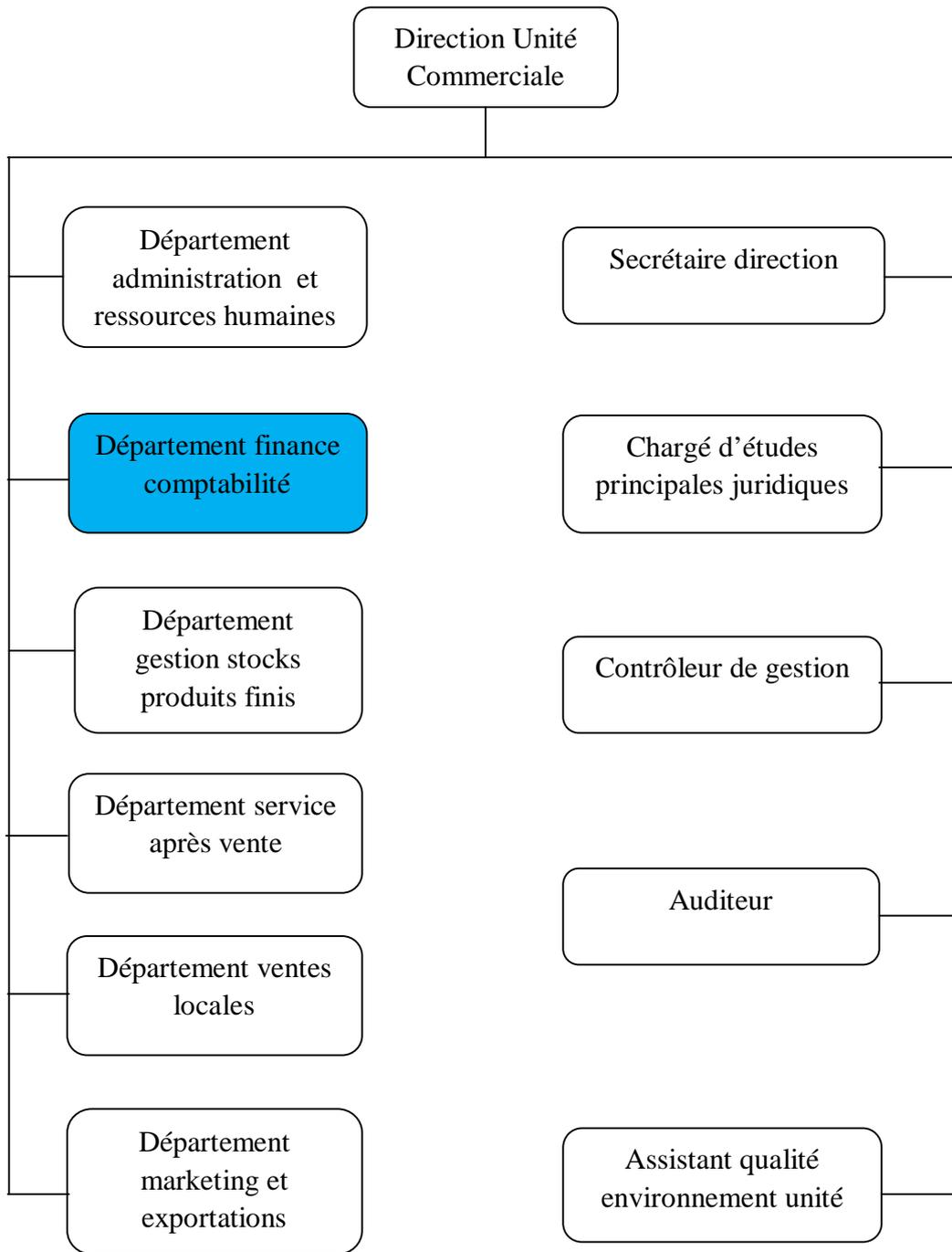


Figure I.2 : organigramme unité commerciale.

L'unité commerciale est composée d'une direction commerciale au sommet de la hiérarchie.

Cette direction a sous sa tutelle six départements qui collaborent pour mettre en œuvre la stratégie commerciale de l'entreprise.

Ces départements sont :

- Ø **Département administration et ressource humaine** : est composé du service de gestion du personnel qui gère les paies et la formation, aussi d'un service moyens généraux qui se charge de la gestion des affaires générales, achats et relations extérieures.

- Ø **Départements finance et comptabilité** : Ce département est composé de deux services : un service comptabilité général et un service comptabilité finance, il assure la comptabilité de toutes les opérations permettant d'établir des bilans et des comptes des résultats ainsi que la gestion du trésorier de l'unité commerciale. La finance concerne la couverture des dépenses des autres départements.

- Ø **Département gestion de stocks des produits finis** : il se compose de deux services :
 - **Service magasins produits finis** : sa fonction est la gestion des stocks de l'entreprise, il doit :
 - Alimenter les stocks sur base de réception de produits.
 - Réintégrer des produits retournés par le client dans le cadre de la garantie ou depuis les stocks extérieur.
 - Réintégrer des produits retournés par le client.
 - **Service programmation** : Ce service assure les livraisons vers les clients par le bien de l'expédition qui ce charge de l'établissement du bulletin de livraison En se basant sur la facture après avoir servit le client. Tendit que la section programmation, se base sur les entrées et sorties établies de manière périodique, la situation des stockes ainsi la balance globale des stockes.

- Ø **Département service après vente** : ce département est responsable de la gestion du circuit SAV (Service Après Vente) à travers le territoire national, plus de 140 agents agréés forment le réseau SAV de l'ENIEM. Ces agents sont à la charge du département service après vente, qui doit les alimenter en pièce de rechange pour assurer les prestations service qui s'inscrivent dans le cadre de la garantie sur les produits de la garantie sur les produits de l'entreprise. Lors des retours des clients, le département SAV à travers une commission technique spécialisée décide de la réparation ou remplacement des produits retournés.

- Ø **Département vente** : ce département se compose de trois services :
 - **Service vente** : charger principalement de :
 - La réception des commandes des clients.
 - La gestion des facturations.
 - La gestion des règlements.
 - Le contrôle des livraisons par rapport aux facturations.
 - **Service client** : enregistre les clients et gère le réseau des agents agréée à travers le territoire nationale.
 - **Service synthèse et recouvrement** : consiste en le recouvrement des créances qui sont générées soit par les ventes à terme, soit par ventes par facilité, les ventes par trait ou par chèque, ainsi que le traitement et le suivi des marchés.

Ø **Département marketing** : consiste à mettre en œuvre la stratégie concurrentielle de l'entreprise, il est chargé aussi de la préparation et l'organisation des salons nationaux.

III. Organigramme département finances & comptabilité (unité commerciale) :

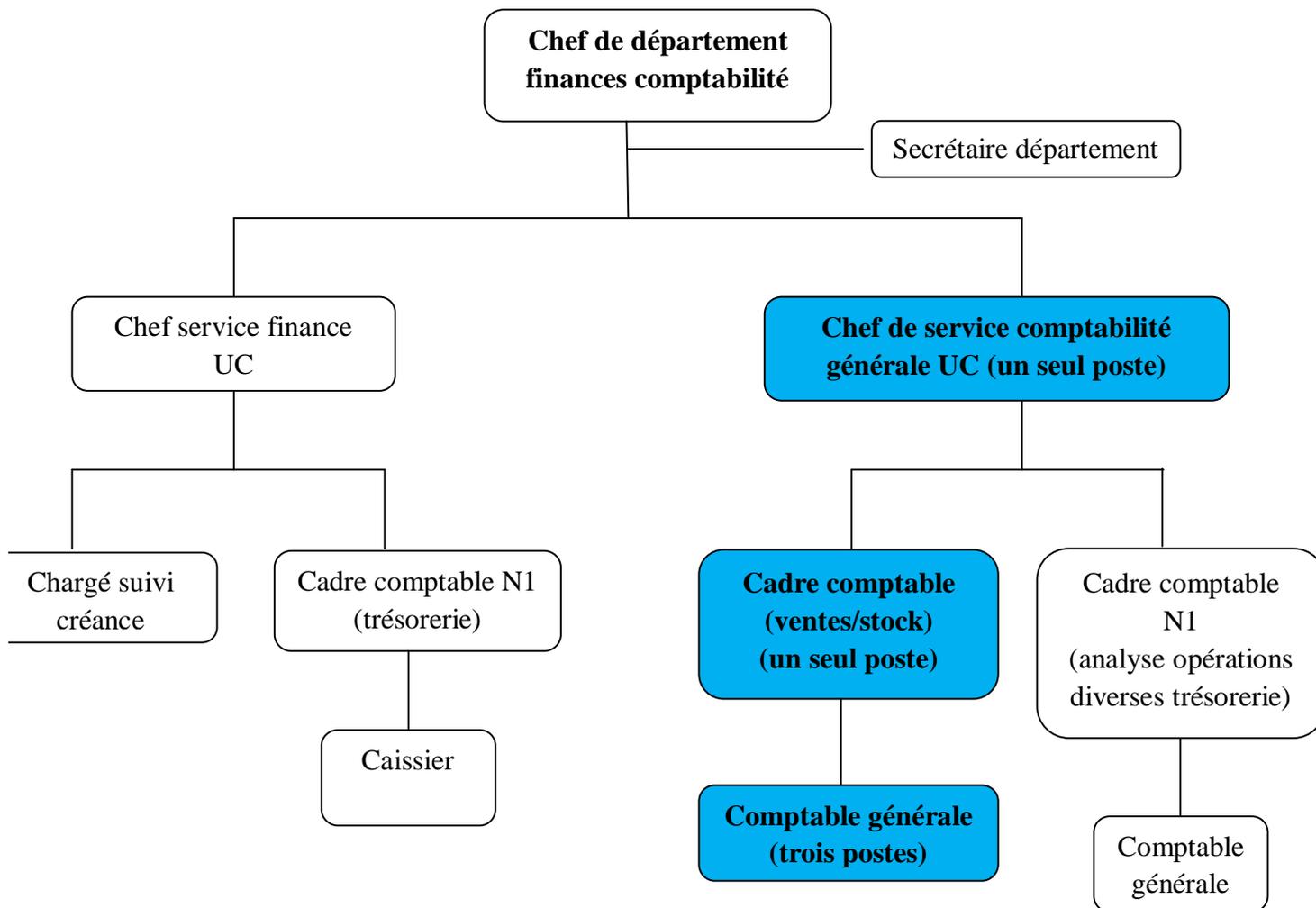


Figure I.3 : organigramme département finance & comptabilité.



Domaine d'étude

UC : Unité Commerciale.

Remarque :

Dans l'organigramme du département finance et comptabilité, on remarque qu'il y a trois postes au niveau du comptable général et cela dans le but d'introduire la notion du **code dossier** qui donnera à chaque comptable général l'accès seulement aux dossiers qui lui ont été attribués.

I. Généralités sur les stocks :**I.1 Définition d'un stock : [B02]**

Le stock, c'est l'ensemble des produits destinés à la vente. C'est aussi l'ensemble des matières premières ou des pièces détachées que possède l'entreprise en vue de fabriquer des produits finis ou de procurer des services. Différentes entreprises ont différentes sortes de stocks. En voici quelques exemples:

- **Pour le fabricant** : le stock c'est l'ensemble des matières premières et des pièces détachées destinées à leur transformation en produits finis puis à la vente. Ces produits finis font également partie du stock
- **Pour le prestataire de service** : Le stock c'est l'ensemble des matériels et produits destinés à fournir un service.

I.2 définition de la gestion des stocks : [B02]

La gestion des stocks est la méthode par laquelle on :

- 1- **Réceptionne le stock** : Comptez et vérifiez l'état des marchandises ou du matériel que vous recevez en fonction du bordereau de livraison.
- 2- **Fait un relevé du stock** : Enregistrez toutes les marchandises entrant ou sortant de votre entreprise.
- 3- **Entrepose le stock** : Gardez le stock qu'il faut dans un endroit pratique et accessible.
- 4- **Arrange le stock** : Arrangez votre stock de façon à attirer vos clients, les inciter à acheter et vous permettre également de le compter.
- 5- **Vérifie le stock** : Vérifiez et comptez souvent votre stock pour vous assurer qu'il est en parfait état et que rien ne manque.
- 6- **Passe une nouvelle commande de stock** : Passez une nouvelle commande en quantité suffisante et au bon moment.

La gestion des stocks est importante. Elle aide à:

- Emmagasiner les marchandises ou matériels adéquats;
- Garder la quantité de stock qu'il faut, c'est-à-dire ni trop ni trop peu;
- Conserver le stock en bon état;
- Eviter des pertes;
- Passer à temps de nouvelles commandes.

Le relevé de stock : [B02]

Faire un relevé de stock signifie enregistrer:

- Toutes les rentrées de stock;
- Toutes les sorties de stock.

Les relevés de stock sont importants parce qu'ils vous disent:

- Quelles marchandises ou quel matériel vous avez vendus ou utilisés;
- Quelle quantité de vos marchandises ou matériel vous avez vendue ou utilisée;
- Quand vous avez vendu des marchandises ou matériel;

- Quelle quantité de marchandises ou matériel vous reste en stock.

Vous pouvez utiliser les informations sur les relevés de stock pour trouver:

- Quel stock se vend vite;
- Quel stock il faut passer encore renouveler;
- Quand il faut passer de nouvelles commandes;
- En quelle quantité il faut passer ces nouvelles commandes;
- Si du stock a disparu.

Lorsque vous faites vos propres relevés de stock, vous pouvez vous servir de fiches, de registres, d'un cahier de feuilles volantes d'un classeur ou de tout autre document convenant à votre entreprise.

Assurez-vous que vos fiches de stock sont à jour et sans erreurs. Un relevé de stock qui n'est pas à jour vous fournit des informations erronées et risque de vous amener à prendre des décisions fausses pour votre entreprise. Trouvez et enregistrez le niveau de commande pour les marchandises que vous vendez beaucoup ou les matériels que vous utilisez beaucoup.

Le niveau de commande est la quantité minimum de chaque matière première ou marchandise que vous jugez nécessaire pour assurer la production ou la vente jusqu'à la livraison du prochain réassortiment. Quand vos stocks atteignent la quantité de sécurité, vous devez passer une nouvelle commande.

L'inventaire : [B02]

L'inventaire est l'exercice qui consiste à compter, puis à enregistrer votre stock sur une fiche d'inventaire.

L'inventaire vous aide à savoir:

- S'il y a du stock en moins et combien;
- S'il y a du stock endommagé ou en mauvais état;
- Quelles marchandises se vendent vite et lesquelles se vendent peu ou pas;
- Quels matériel et pièces détachées vous utilisez le plus souvent et lesquels vous utilisez le moins;
- Quand il faut repasser des commandes si vous ne gardez pas de fiches de stock.

Suivez les étapes ci-dessous pour faire un inventaire:

1. Assurez-vous que votre stock est bien arrangé.
2. Préparez votre fiche d'inventaire.
3. Comptez le stock et enregistrez la quantité sur les fiches d'inventaire.
4. Recopiez les informations de votre fiche de stock sur votre liste d'inventaire.
5. Comparez votre fiche de stock avec votre liste d'inventaire.
6. Ecrivez les quantités correctes sur votre fiche de stock.

Utilisez les informations tirées de votre inventaire pour améliorer votre gestion des stocks.

II. Nécessité d'une bonne gestion des stocks : [B02]

Si les capitaux immobilisés constituent la structure de l'entreprise, les valeurs d'exploitation (les stocks) lui assurent tout son dynamisme et réalisent le but même de l'entreprise (acheter pour revendre en réalisant un profit). Le "bien-être" de l'entreprise exige une rotation rapide de ces valeurs entre différents postes.

Les nombreux changements de situation que subit ce groupe de l'actif engendrent la nécessité :

Au point de vue comptable:

- de noter et d'analyser en profondeur l'évolution des comptes de stocks.
- de dresser et d'interpréter des situations changeantes.

Au point de vue extra-comptable : (c'est à dire en dehors de la comptabilité) :

De fournir à la direction une multitude d'informations indispensables à la gestion de l'entreprise.

III. La comptabilisation des stocks : [B03]

La comptabilisation de stock doit décrire en quantité et en valeur tous les mouvements articles en stock.

Elle s'établit à partir d'un certain nombre de documents réglementaires fournis selon une périodicité variable.

La comptabilisation de stock est tenue par le responsable des services économiques, qui enregistre les mouvements des articles stockés dans l'établissement en quantité et en valeur sur les registres suivants :

- **Le journal des stocks** : [W04]
C'est un document comptable qui enregistre de façon chronologique toutes les opérations effectuées par l'entreprise.
- **Le grand livre des stocks** : [W05]
Est un livre comptable où figure l'inventaire des articles qu'une entité détient dans le but de les vendre ou de les utiliser pour fabriquer un produit ou rendre un service. Il contient tous les comptes d'actif, de passif, de capitaux propres, de produits et de charge de l'entité.

Ces documents sont conçus pour être servi simultanément par écriture unique en faisant usage de la duplication ou par autre support de caractère informatique.

- **La balance comptable** : [B02]
Est un document comptable qui affiche pour chaque produit les quantités et valeurs pour tout les mouvements d'entrées et de sorties pour une période donnée. Elle est aussi d'un apport crucial de contrôle des produits et facilite l'audit.

III.1 Aspect comptable de la gestion des stocks : [B06]**1. L'inventaire permanent :****a) Définition :**

En comptabilité générale, on pratique, la plupart du temps, un inventaire extra-comptable, matériels (physique) au moins une fois par an. Il s'agit de l'inventaire intermittent. L'enregistrement des stocks en permanence n'est pas jugée utile à l'organisation.

En comptabilité de gestion, il est nécessaire de connaître à tout instant la valeur des éléments en stock car ils contribuent à la formation des coûts mais aussi à connaître la valeur des existants en fin de période (stock final).

b) Sortes d'inventaire permanent :**• L'inventaire permanent en quantité :**

Il importe de mettre en place une procédure de gestion (de suivi) physique des stocks.

Pour cela, nous pouvons utiliser un système à base de bons : bon d'entrée (bon de réception), bon de sortie avec visa du magasinier, bon de retour au fournisseur, bon de réintégration... Une fiche de stock doit être établie par article comprenant la date d'entrée, de sortie et de stock final.

Cette gestion nécessite la mise en place d'un système informatisé, d'où l'importance d'un bon outil informatique (codes barres,...).

• L'inventaire permanent en valeur :

Une fois le dénombrement réalisé, la phase suivante consiste à évaluer le stock. Conformément au principe du coût historique, chaque article en stock doit être évalué au coût d'entrée, c'est-à-dire au coût de revient total des charges engagées pour amener l'article en question à l'état où il se trouve en stock. Selon la nature du stock, ce coût de revient est le coût d'acquisition ou le coût de production.

2. La valorisation des entrées et des sorties :**a) La valorisation des entrées :**

Elle s'effectue au coût d'achat ou au coût d'acquisition. Le coût d'achat peut être défini comme étant tout ce qu'a coûté le bien jusqu'à sa mise en stock.

Les éléments achetés tels que les marchandises, matières premières, emballage, et d'autres approvisionnements seront valorisés au coût d'achat.

Les éléments produits (produits finis, produits intermédiaires, en cours...) sont, eux, valorisés au coût de production.

b) La valorisation des sorties :

Elle permet d'une part, de mesurer la contribution des éléments sortis du stock à la formation des coûts, et, d'autre part, d'apprécier la nature des existants, c'est-à-dire la valeur du stock final.

Le principe de la valorisation des sorties est le suivant :

Tout élément qui sort du stock, sort au coût auquel il est entré.

Ce principe n'est pertinent que pour les éléments parfaitement individualisables (exemple : œuvre d'art). Généralement, les éléments en stock sont les mêmes, ils sont fongibles et non individualisables, par ailleurs ils ont été achetés ou produits à des dates différents, à des coûts différents qui peuvent varier en fonction des

quantités. Ainsi, afin de pallier à cet inconvénient, on utilise des méthodes de détermination de la valeur des stocks qui s'approchent de la réalité des coûts telle que les méthodes d'épuisement des lots et les méthodes du coût unitaire moyen pondéré.

3. Les méthodes d'épuisement des stocks : [B06]

a) Méthode F.I.F.O (first in, first out):

Méthode du "**premier entré, premier sorti**" fondée sur le principe que les marchandises sortent du magasin dans leur ordre d'entrée. Automatiquement, le stock sera valorisé au prix des derniers achats effectués.

Cette méthode est avantageuse en période de baisse de prix car on valorise les sorties au plus ancien prix payé qui est le plus élevé; ainsi, les stocks comptables seront valorisés au prix le plus récent, c'est-à-dire le plus bas.

Automatiquement, le stock sera valorisé au prix des derniers achats effectués.

b) Méthode L.I.F.O. (Last in, first out):

Méthode du "**dernier entré, premier sorti**", fondée sur le principe que les prix les plus récents doivent être opposés aux prix de vente, le stock étant valorisé au prix des achats les plus anciens.

C'est une méthode avantageuse quand les prix sont en hausse car les sorties sont valorisées aux derniers prix payés qui sont les prix les plus élevés et le stock comptable est valorisé aux prix des stocks les plus anciens, c'est-à-dire les plus bas.

Le choix entre FIFO et LIFO dépend de la tendance de l'évolution des prix et vise à valoriser le stock comptable au prix le plus bas.

4. Les méthodes du coût unitaire moyen pondéré (CUMP) : [B06]

Le **coût unitaire moyen pondéré (CUMP)** est une valorisation des sorties d'un poste comptable à un coût unitaire qui est la moyenne des coûts unitaires des entrées de la période (en y incluant le coût unitaire du stock initial). Cette moyenne des coûts unitaires est pondérée par les quantités.

a) **La méthode du CUMP Fin de période avec stock initial :**

La valorisation des sorties des stocks s'effectue à la fin de la période de référence avec le stock initial. Nous ne pouvons valoriser les sorties tant que la période de référence n'est pas terminée.

La période de référence choisie dépend de la rotation des stocks (semaine, mois, trimestre, semestre, année).

$$CUMP = \frac{\text{valeur (stock initial + entrées)}}{\text{quantités (stock initial + entrées)}}$$

b) La méthode du CUMP après chaque entrée :

La méthode CUMP est déterminée après chaque entrée entre deux entrées, toutes les sorties sont évaluées au dernier CUMP.

$$CUMP = \frac{\text{valeur (stock + entrée)}}{\text{quantités (stock + entrée)}}$$

c) Le CUMP Fin de période sans stock initial :

La valorisation des stocks s'effectue à la valeur moyenne des achats de la période.

$$CUMP = \frac{\text{valeur (entrées)}}{\text{quantités (entrées)}}$$

5. L'écriture comptable : [B03]**Définition :**

Une écriture comptable est une opération consistant à enregistrer un flux commercial, économique ou financier à l'intérieur de comptes. Les écritures sont portées dans un document appelé « journal ».

A la fin de l'exercice comptable, l'entreprise doit enregistrer dans sa comptabilité la valeur de ses stocks à la clôture des comptes, sur la base de l'inventaire physique réalisé. Si elle fonctionne tout au long de l'année en inventaire permanent, elle n'aura qu'à ajuster éventuellement la valeur du stock calculé avec le stock réel constaté.

En inventaire intermittent, les stocks ne sont mouvementés du côté comptable que lors des écritures comptables d'inventaire, par une double imputation, l'annulation du stock initial (de début d'exercice) et la constatation du stock final (de fin d'exercice).

Conclusion :

Dans ce premier chapitre nous avons pu situer le projet dans son cadre général en présentant l'organisme d'accueil et ses activités. Dans le chapitre suivant, nous allons procéder à une étude détaillée de l'existant pour dégager ses limites.

Introduction :

Avant d'entamer l'élaboration de notre application, nous avons jugé primordial de présenter les objectifs d'une telle application à partir des éléments moteurs par lesquels elle est constituée.

I. Description de la procédure de travail :

Notre projet consiste à la mise en place d'une application multi-dossier pour la comptabilisation des stocks des produits finis, ces informations sont transmises du département de gestion des stocks vers le département comptabilité, elles consistent en tous les mouvements des stocks (livraison, réception, reconversion...) que le département comptabilité saisit ses informations à leur niveau pour compléter le bilan fiscal d'entreprise et il sera en mesure de communiquer la valeur en stock selon les différents type de déstockage (FIFO, LIFO, CUMP). Dans le but d'aboutir à des résultats similaires à ceux du département de gestion des stocks lors de chaque inventaire.

Remarque :

Multi-dossier : avoir plusieurs bases de données en une seule codifiée par code-dossier.

II. Positionnement du problème :

La comptabilisation des stocks de l'entreprise ENIEM exige un travail fastidieux qui demande un temps considérable en particulier les gestionnaires qui se trouvent confrontés à des problèmes dans l'exécution des activités relatives à la tenu du stock.

Après notre rencontre avec les responsables nous avons recensé leurs problèmes tel que :

- Les tâches fastidieuses et ordinaires (calcul de la balance comptable, CUMP ...).
- La perte d'information (perte des documents) qui se trouvent au niveau des services.
- Perte de temps en remplissant les documents des entrées et sorties des produits.
- Le non tenu à jour des fichiers (mise à jour du registre « mouvement journalier», « facture soldée »).

III. Objectif de l'étude :

La comptabilisation des stocks est un ensemble d'opérations complexes et fastidieuses, pour cela en mettant en place un nouveau système performant qui permettra d'atteindre des objectifs qui sont résumés comme suit :

- Gagner du temps dans les traitements des données.
- Accélérer la sortie des résultats.
- Maitriser les stocks c'est-à-dire connaître le stock, ses mouvements (entrées, sorties) à tout moment, pouvoir l'évaluer fiablement.

Car la maîtrise des stocks peut avoir des répercussions (négative notamment) sur la planification de la production ainsi que sur les ventes.

- Permettre la mise à jour régulière des entrées et sorties.

- Eviter la perte d'informations
- Edition des états (état d'entrée, état de sortie ...)
- D'automatiser les différentes procédures de gestion de la comptabilité des stocks (la réception, la réintégration...).
- Gain du temps considérable dans la recherche des informations (la liste des produits réceptionnés à une date précise) et dans la réalisation des différentes procédures.

Conclusion :

Dans ce chapitre nous avons présenté théoriquement notre sujet pour mieux comprendre la situation existante et de cerner les procédures relative à la comptabilité des stocks des produits finis. Ce qui nous a permis de localiser les problèmes et de déduire les anomalies dont elle souffre.

Ainsi, nous avons dégagé et présenté de manière globale une solution informatique dans cette partie, et nous allons la détailler dans la partie qui suit.

Introduction :

La conception de toute solution est d'une grande importance et elle doit être traitée avec rigueur et précision, car elle constitue la base du système à développer.

Avant de s'engager dans la conception, il est impératif de passer par la phase d'analyse qui permet d'identifier les différents acteurs qui interagissent avec le système ainsi que leurs besoins. Puis on passe à la conception qui, en s'appuyant sur les résultats de la phase d'analyse, donnera la description détaillée du système cible et des objectifs à atteindre.

Dans ce chapitre, nous étudions dans un premier temps les besoins de notre système, ensuite, une spécification formelle des besoins est présentée par des diagrammes de cas d'utilisations et de séquences suivant la modélisation UML.

I. Identification des acteurs :

I.1. Définition d'un acteur : [B07]

Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation. Ainsi les utilisateurs d'un système appartiennent à une ou plusieurs classes d'acteurs selon les rôles qu'ils tiennent par rapport au système.

Une même personne physique peut se comporter en autant d'acteurs différents que le nombre de rôles qu'elle joue vis-à-vis du système.

Ainsi par exemple, l'administrateur d'un système de messagerie peut être aussi utilisateur de cette même messagerie. Il sera considéré, en tant qu'acteur du système, dans le rôle d'administrateur d'une part et dans celui de l'utilisateur d'autre part.

Un acteur peut aussi être un système externe avec lequel le cas d'utilisation va interagir.

I.2. Formalisme graphique : [B07]



Nom de l'acteur

I.3 Identification des acteurs : [B12]

Dans le cas de notre application, nous avons trois acteurs qui interagissent avec le système :

- Ø **Comptable générale :** traite, conformément à la réglementation en vigueur (contrôle, imputation comptable, confection de pièces...) des pièces et des documents reçus, relatifs aux opérations d'achats, exploitation et comptabilisation au jour sur le système informatique, afin d'obtenir, dans les délais, des situations sincères et complètes.
- Ø **Cadre comptable :** coordonne, dirige, surveille et contrôle l'enregistrement comptable au jour le jour des opérations et des achats en vue d'obtenir des états de synthèses conformes.
- Ø **Chef de service comptabilité générale UC :** assure l'enregistrement au jour le jour des flux comptables conformément au plan comptable national et à la réglementation en vigueur et ce, pour obtenir, dans les délais, des états de synthèses conformes, sincères et véritables en

vue de permettre aux responsables de l'Unité de prendre des décisions de gestion qui s'imposent.

II. Identification des cas d'utilisation :

II.1. Définition : [B07]

Un cas d'utilisation correspond à un certain nombre d'action que le système devra exécuter en réponse à un besoin d'un acteur. Un cas d'utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou partie prenante du système.

Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation.

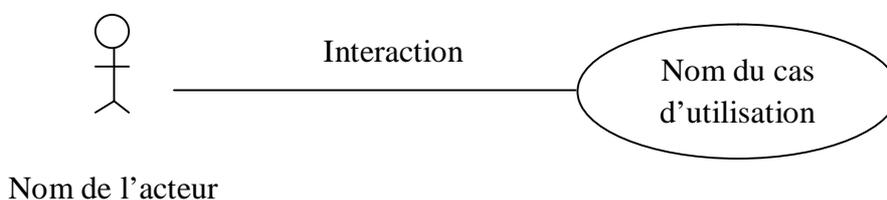
II.2. Formalisme graphique : [B07]

Un cas d'utilisation se représente par un ovale dans lequel figure son intitulé.

L'interaction entre un acteur et un cas d'utilisation se représente comme une association.

Elle peut comporter des multiplicités comme toute association entre classes.

Le formalisme de base de représentation d'un cas d'utilisation est le suivant :



Le tableau suivant récapitule les cas d'utilisation de chaque acteur de notre application :

Acteur	Tâches
Comptable générale	-Se connecter -S'authentifier -Accéder à l'espace dossier -Accéder aux fichiers Ø Fournisseur Ø Client Ø Article Ø Dépôt - Accéder à l'édition Ø Liste des entrées Ø Correction des stocks physiques Ø Liste des sorties - Accéder aux états Ø Mouvement des entrées Ø Mouvement sorties Ø Prix moyen pondéré Ø LIFO Ø FIFO -Se déconnecter
Cadre comptable	-Idem que comptable générale -Saisie d'inventaire -Contrôle des écarts

	-Consultation des états -consolidation
Chef de service comptabilité générale UC	-Idem que le cadre comptable -Contrôle de l'inventaire - Contrôle des Entrées/Sorties

Tableau III.1 : identification des cas d'utilisation.

Remarque :

- a. **Espace dossier** : image ou environnement de l'application.
- b. **Code dossier** : c'est un code que l'utilisateur saisit pour accéder à l'espace dossier. Cela va lui permettre seulement d'accéder à un dossier attribué.
- c. **Etat** : est une mise en forme (disposition, calcul) d'informations extraites d'une base de données, en vue de leur affichage ou de leur impression.
- d. **Consolidation** : est le regroupement partiel ou total de données de l'espace dossier. Elle concerne des données organisées ou liées entre elles. Plus spécifiquement il s'agit d'un regroupement cumulé des données sous forme d'un tableau.

III. Les cas d'utilisation :

Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré dont une description textuelle sera présentée.

Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un diagramme d'activité.

La figure suivante donne la représentation graphique de la démarche de modélisation adoptée pour concevoir l'application :

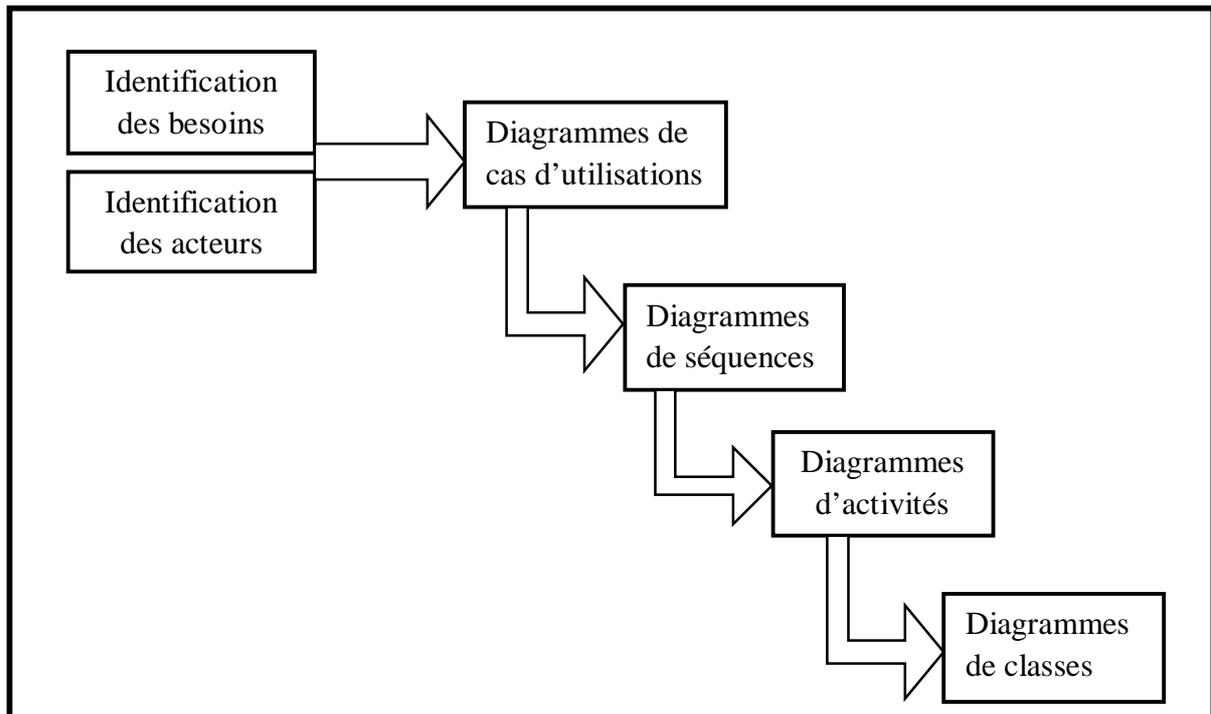


Figure III.1 : La démarche adoptée pour la modélisation

III.1 Diagrammes de cas d'utilisation :

Cas d'utilisation du comptable général :

Cas d'utilisation du cadre comptable:

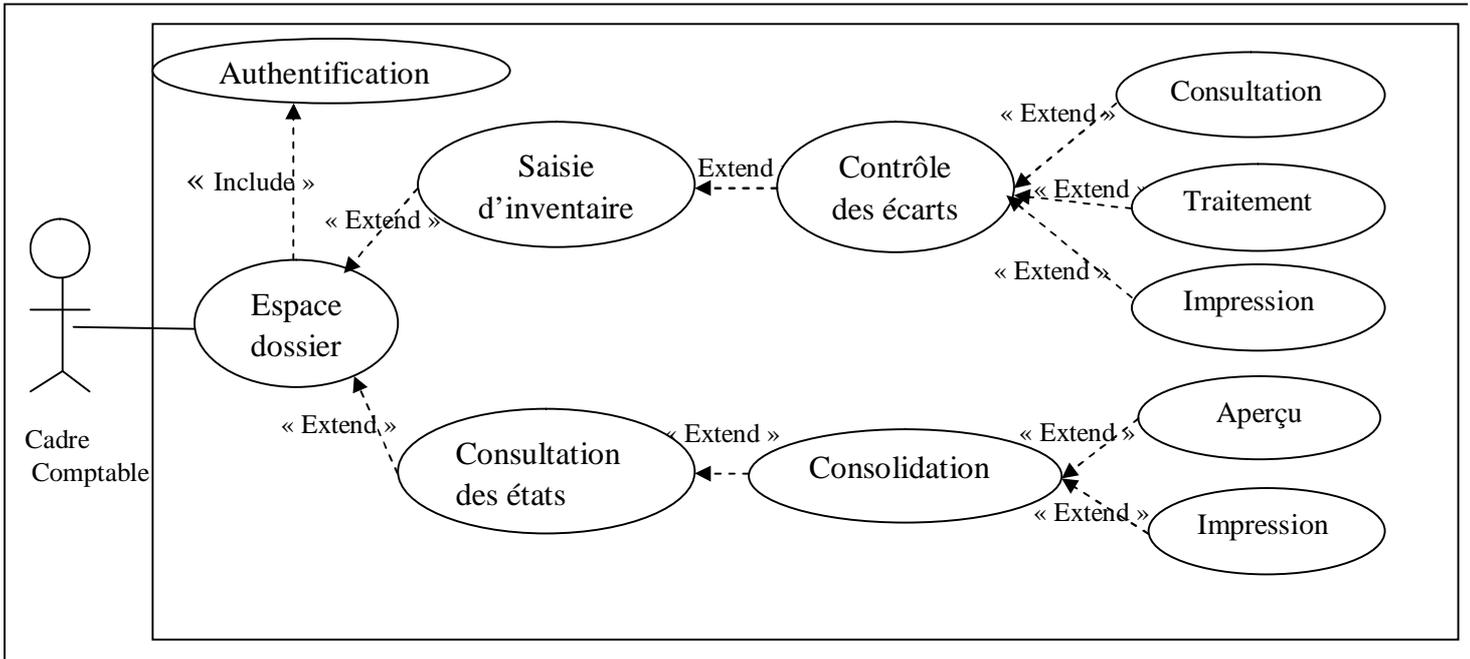


Figure III.3 : Cas d'utilisation du cadre comptable.

Cas d'utilisation du chef de service:

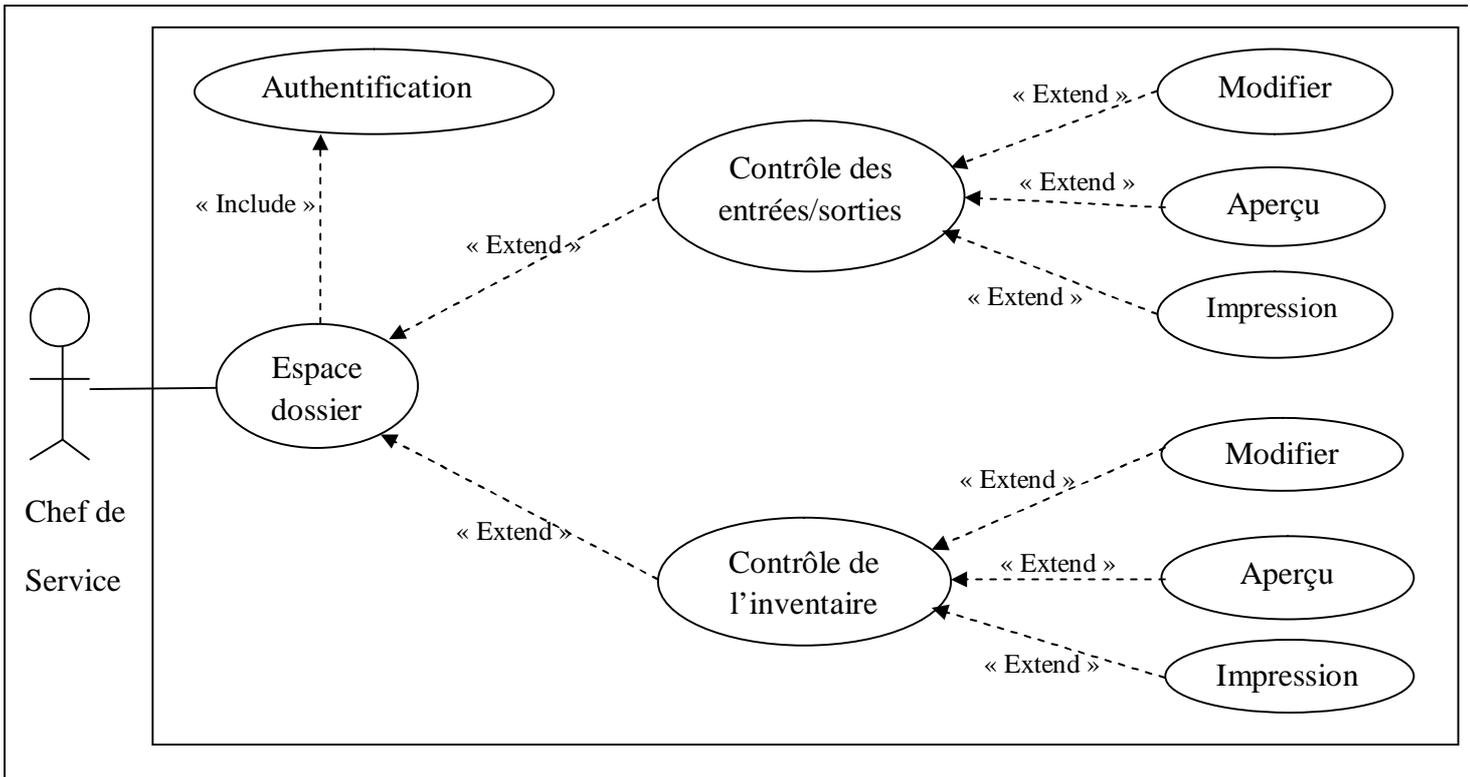


Figure III.3 : Cas d'utilisation du chef de service comptabilité.

III.2 Description des cas d'utilisation avec des scénarios :

Pour détailler le déroulement d'un cas d'utilisation, la procédure la plus évidente consiste à recenser la façon textuelle toutes les interactions entre les acteurs et le système.

Dans ce qui suit nous décrivons donc quelques cas d'utilisation de notre système.

Cas d'utilisation : S'authentifier.

Acteur : comptable général.

Résumé : ce cas d'utilisation permet au comptable général d'accéder aux dossiers qui lui ont été attribués et de s'authentifier.

Enchaînement : ce cas commence lorsque le comptable général accède à la page principale de l'application.

Scénario nominal :

1. Le comptable général saisie le login, le mot de passe et le code dossier puis clique sur « connexion ».
2. Le système vérifie l'existence du dossier et conformité du login et du mot de passe.
3. Le système affiche la page principale de l'espace dossier.

Scénario alternatif :

Le code dossier et/ou le login et/ou mot de passe saisie par le comptable générale sont incorrectes.

L'enchaînement commence au point 2 du scénario nominal.

3. Le système affiche un message d'erreur sur le formulaire d'authentification demandant au comptable de ressaisir le code dossier et/ou le login et/ou le mot de passe.

Le scénario nominal reprend au point 1.

Figure III.5 : Description du cas d'utilisation « S'authentifier »

Cas d'utilisation : Ajouter une entrée

Acteur : Comptable général

Résumé : Ce cas d'utilisation permet d'ajouter une entrée.

Enchaînement :

Ce cas d'utilisation commence lorsque le comptable général clique sur « Ajout entrée ».

Scénario nominal :

1. Le système affiche le formulaire de renseignement.
2. Le comptable général remplit le formulaire et clique sur « valider ».
3. Le système vérifie la validité des données saisies.
4. Le système affiche un message de confirmation.

Scénario alternatif :

Le comptable général a saisi une donnée invalide.

L'enchaînement commence au point 3 du scénario nominal.

4. Le système affiche un message d'erreur indiquant la donnée invalide.

Le scénario reprend au point 2 du scénario nominal.

Figure III.6 : Description du cas d'utilisation « Ajouter une entrée»

Cas d'utilisation : supprimer une sortie.

Acteur : comptable général.

Résumé : Ce cas d'utilisation permet de supprimer une sortie.

Enchaînement : Ce cas d'utilisation commence lorsque le comptable général clique sur « supprimer une sortie ».

Scénario nominal :

1. Le système affiche une liste de toutes les sorties.
2. Le comptable sélectionne la sortie puis clique sur « supprimer ».
3. Le système affiche un message de confirmation de la suppression.

Figure III.7 : Description du cas d'utilisation «supprimer une sortie»

Cas d'utilisation : ajouter un article

Acteur : comptable général.

Résumé : Ce cas d'utilisation permet d'ajouter un article.

Enchaînement : Ce cas d'utilisation commence lorsque le comptable général clique sur « Créer ».

Scénario nominal :

1. Le système affiche le formulaire de renseignement.
2. Le comptable général remplit le formulaire puis clique sur le bouton « ajouter ».
3. Le système vérifie la validité des données saisies.
4. Le système crée un nouvel article et affiche un message de confirmation.

Scénario alternatif : le comptable générale a saisi une donnée invalide.

L'enchaînement commence au point 3 du scénario nominal.

4. Le système affiche un message d'erreur sur le formulaire indiquant la donnée invalide.

Le scénario nominal reprend au point 2 .

Figure III.8 : Description du cas d'utilisation «Ajouter un article»

IV. Diagrammes de séquences et diagrammes d'activités :

Après avoir décrit les cas d'utilisation avec des scénarios, nous allons traduire ces derniers en diagrammes de séquences et en diagrammes d'activités.

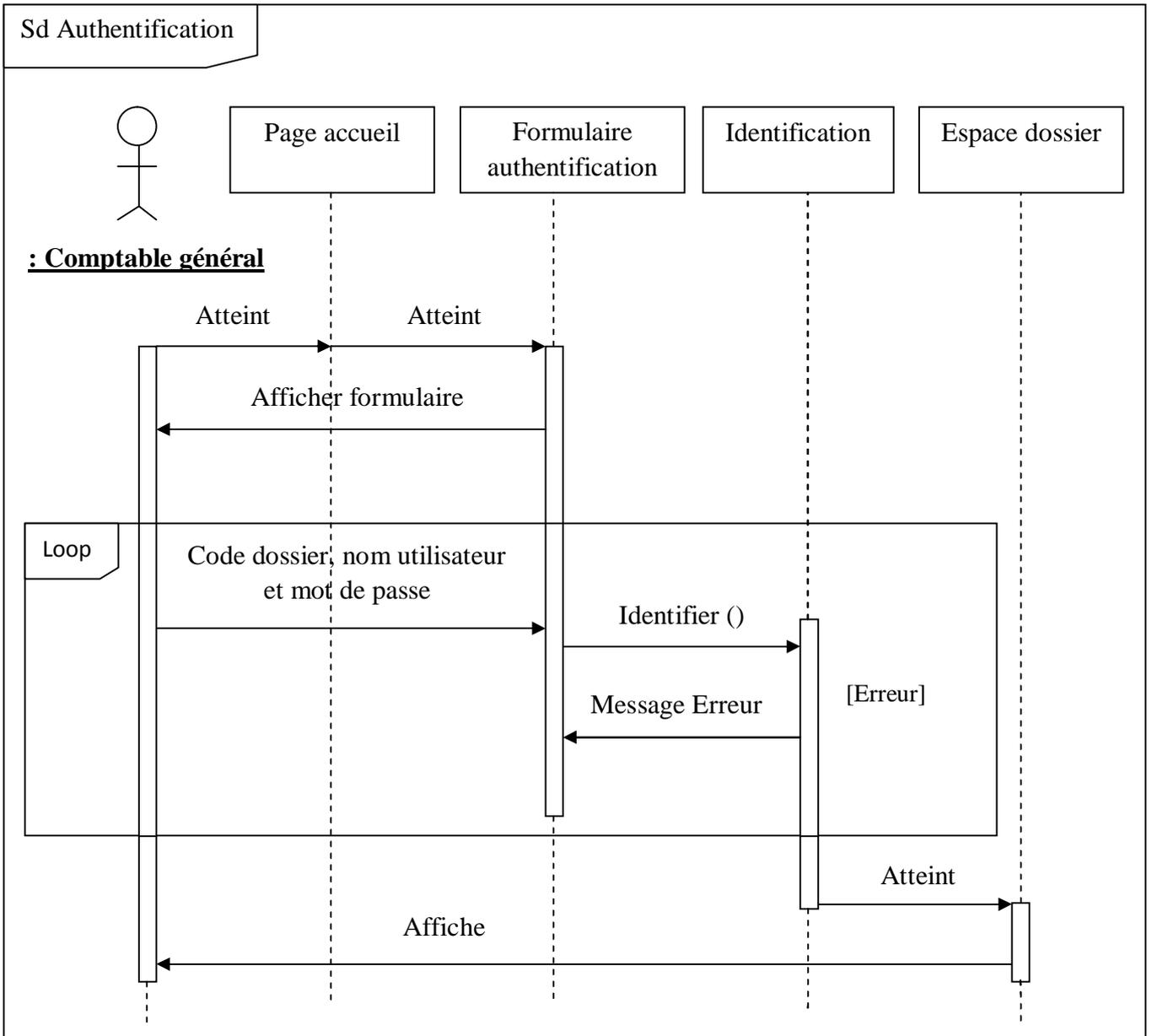


Figure III.9 : Diagramme de séquence du cas d'utilisation

« S'authentifier ».

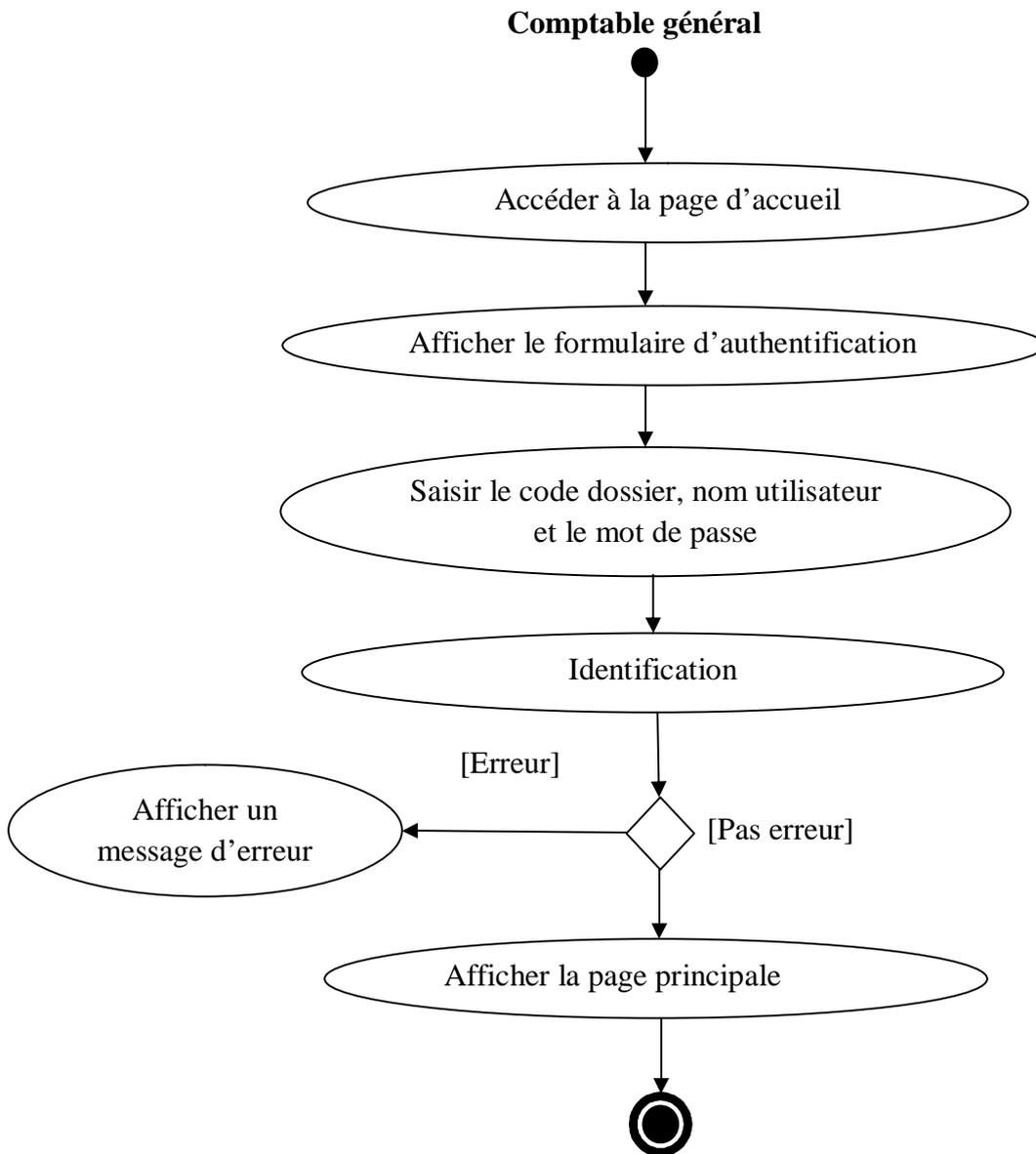


Figure III.10 : Diagramme d'activités du cas d'utilisation « S'authentifier ».

Exemple :

Supposons que le comptable saisit son code dossier qui est « Annaba », son login et mot de passe, dans ce cas il aura accès à ce dossier ainsi qu'à la consultation des dossiers qui ne lui ont pas été attribués à travers le dossier consolidé.

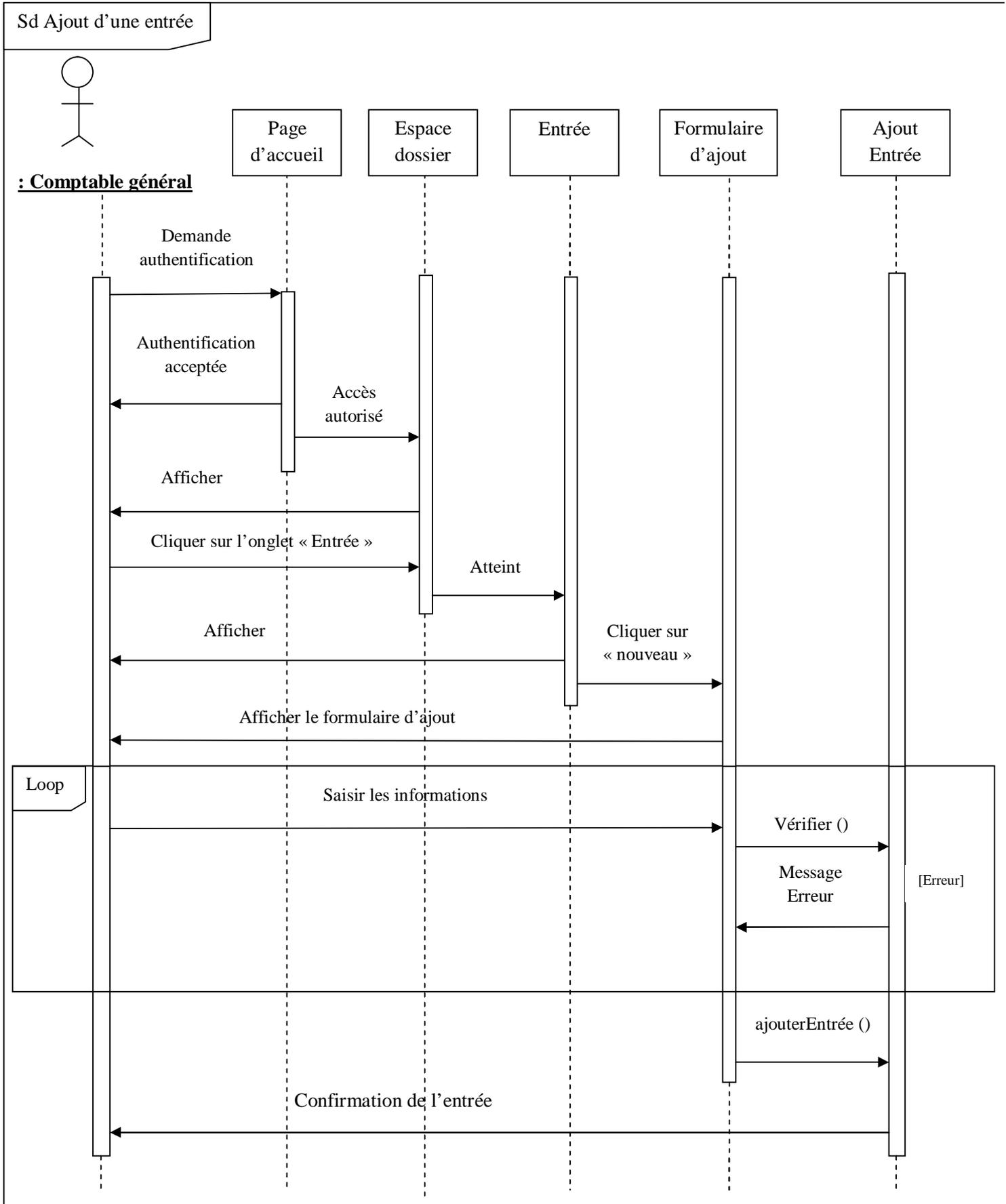


Figure III.11 : Diagramme de séquence du cas d'utilisation « Ajout d'une Entrée».

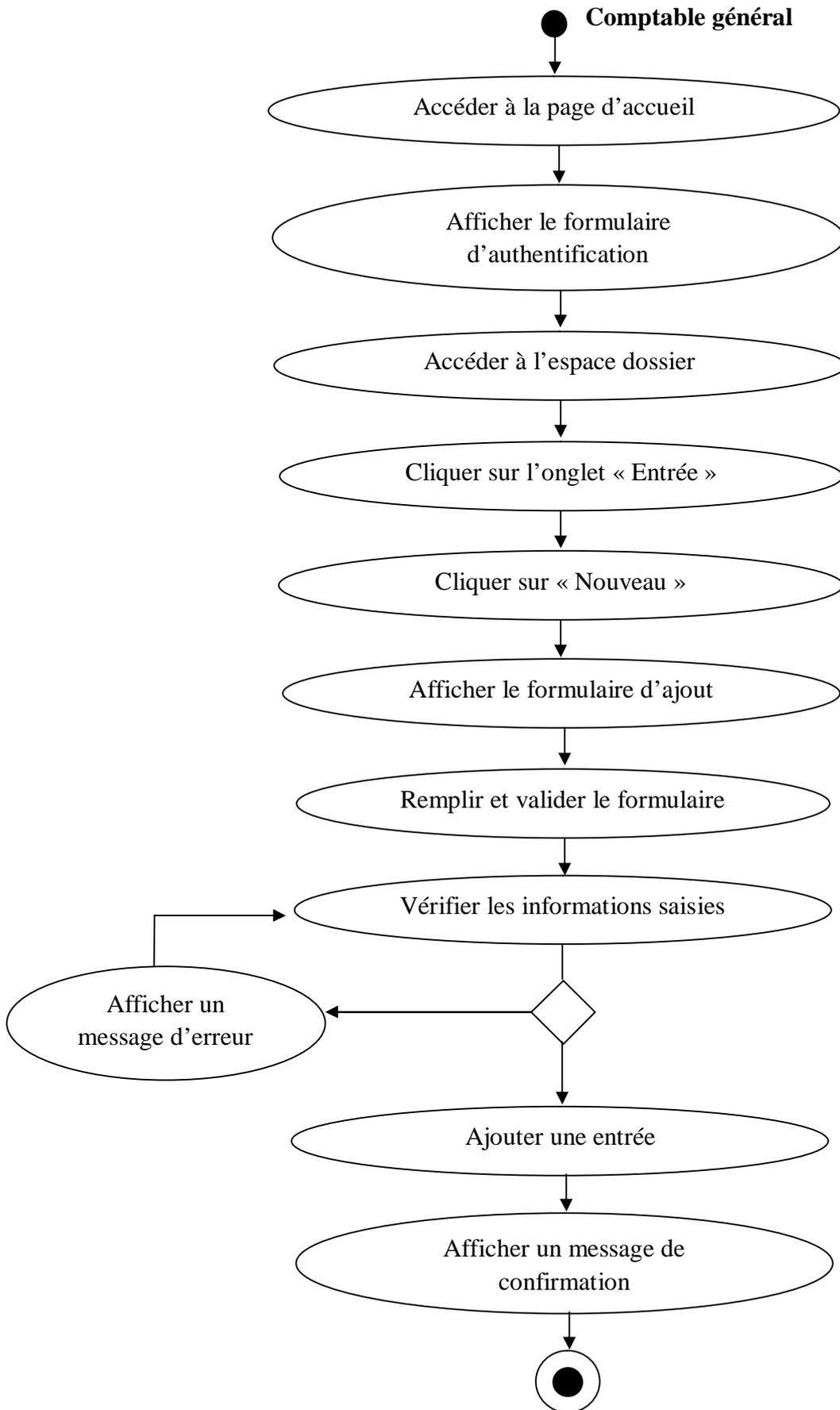


Figure III.12 : Diagramme d'activité du cas d'utilisation « Ajout d'une entrée »

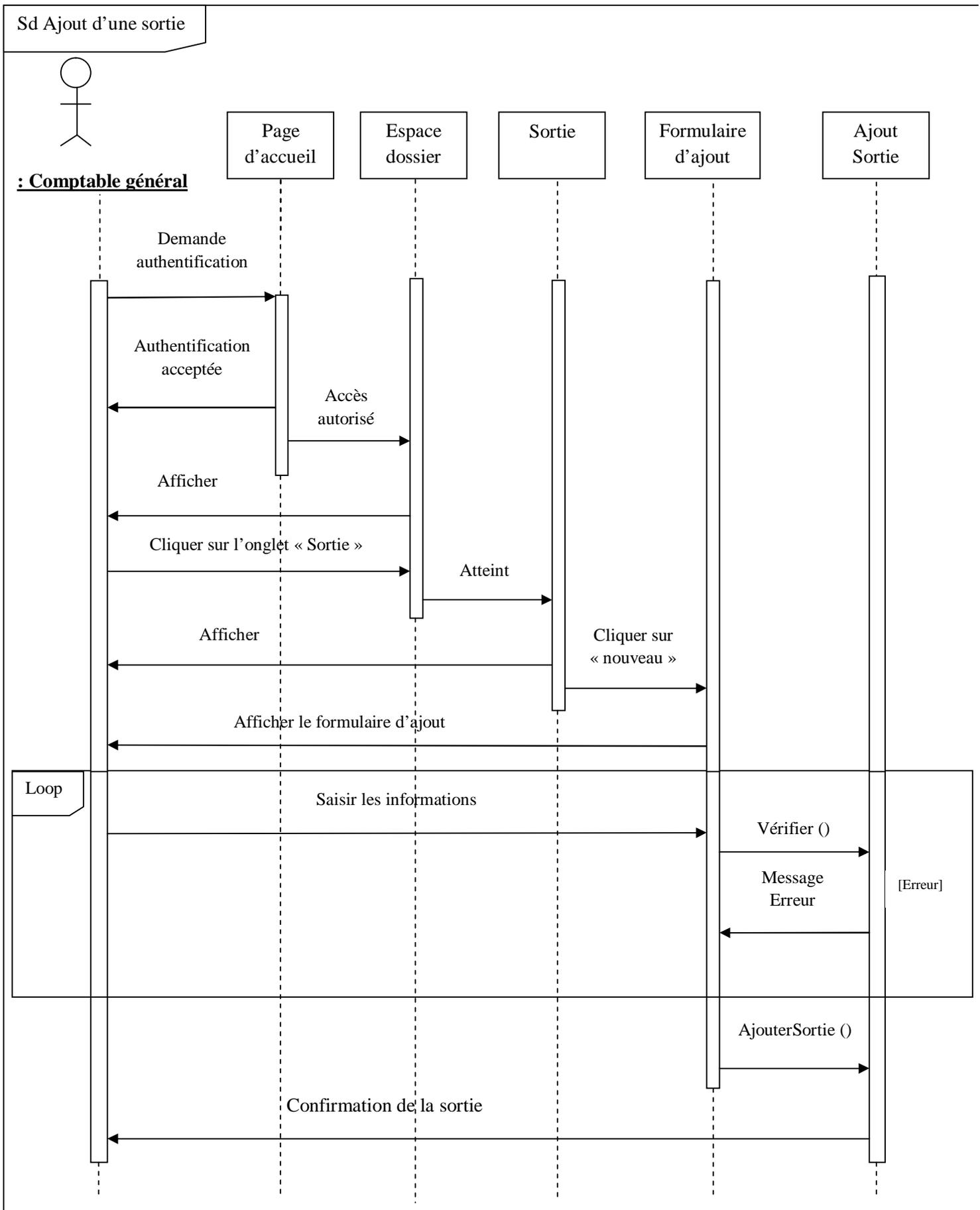


Figure III.13 : Diagramme de séquence du cas d'utilisation « ajout d'une sortie ».

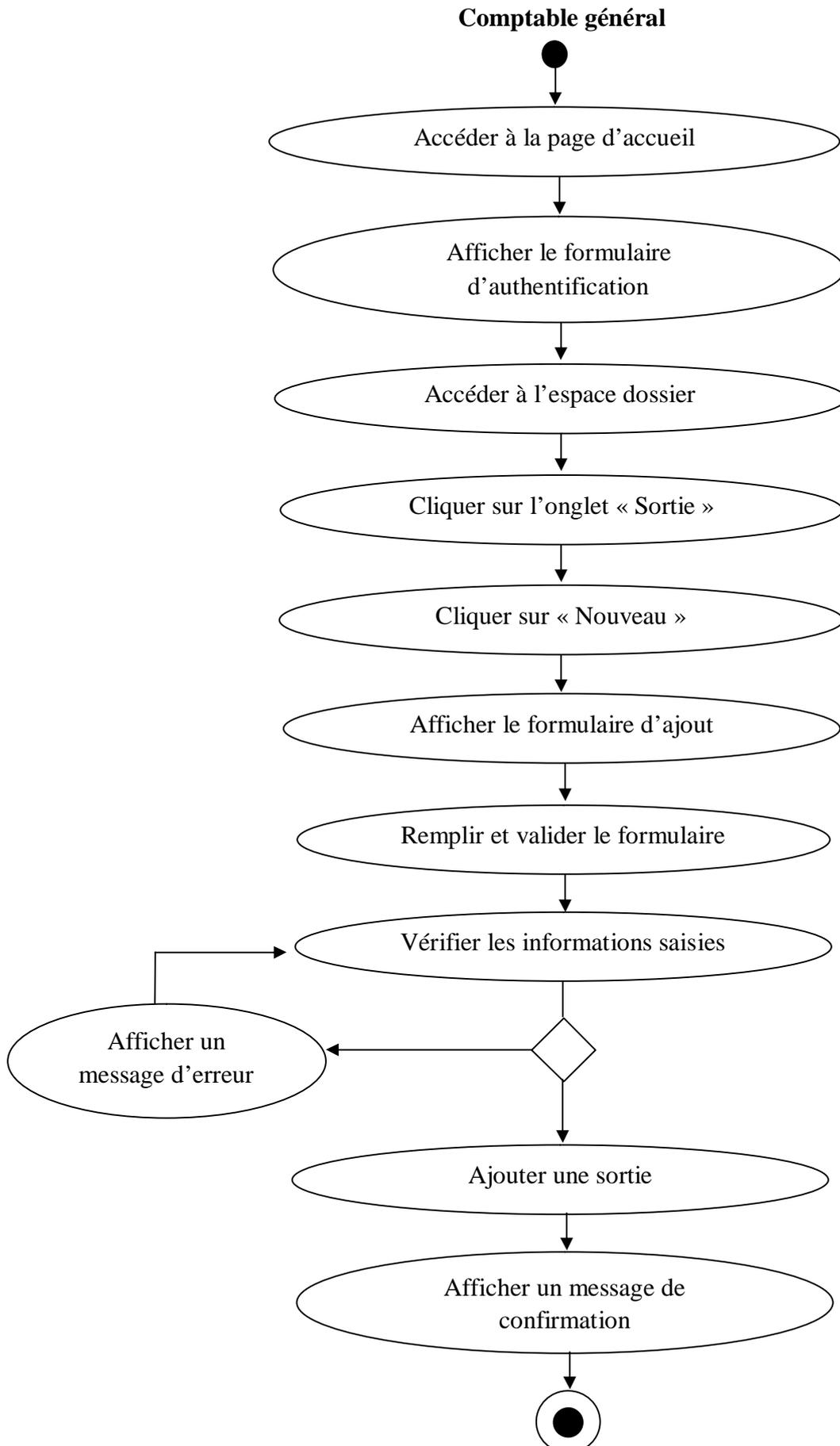


Figure III.14 : Diagramme d'activité des cas d'utilisations « ajout d'une sortie ».

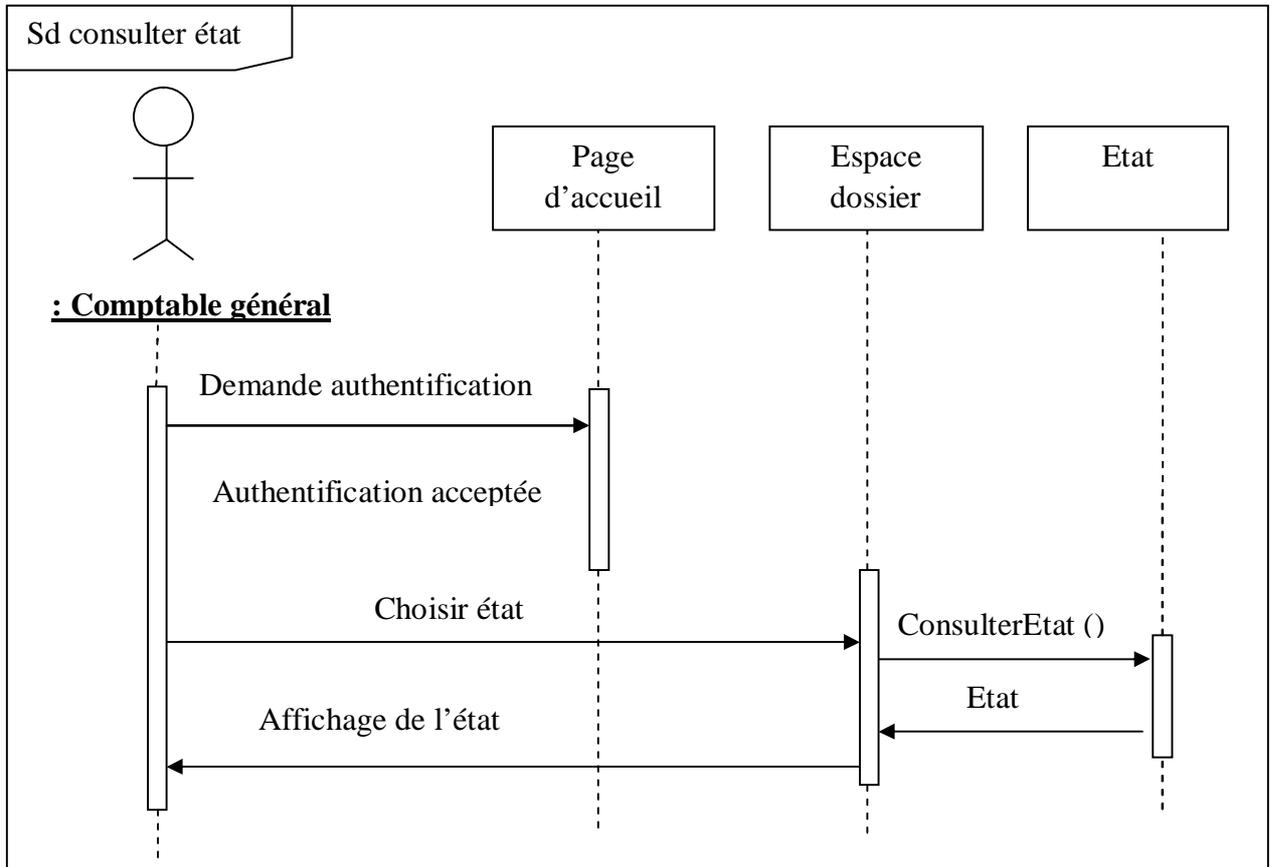


Figure III.15 : Diagramme de séquence pour le cas d'utilisation « consultation des états ».

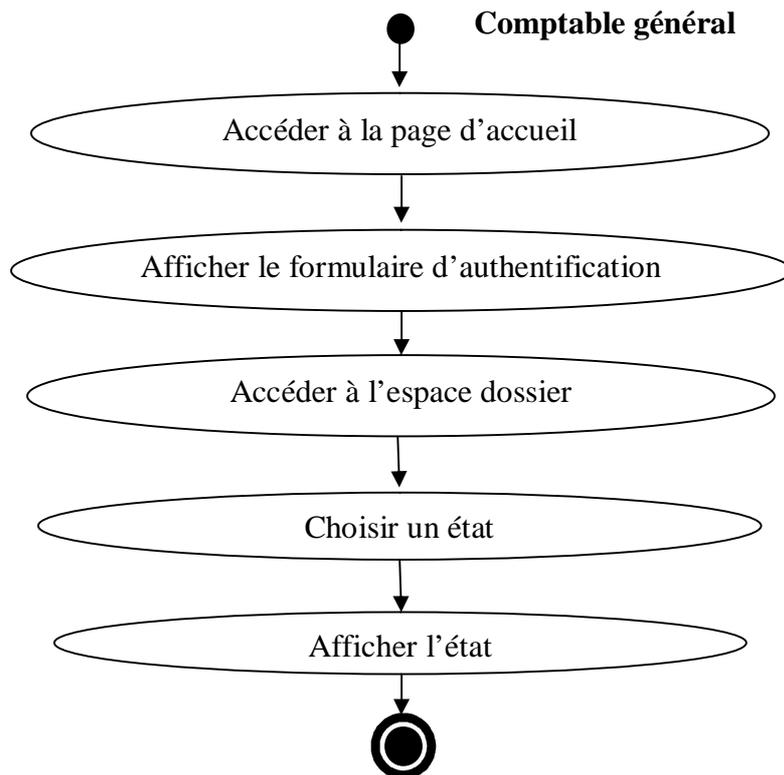


Figure III.16 : Diagramme d'activité du cas d'utilisation « consultation des états ».

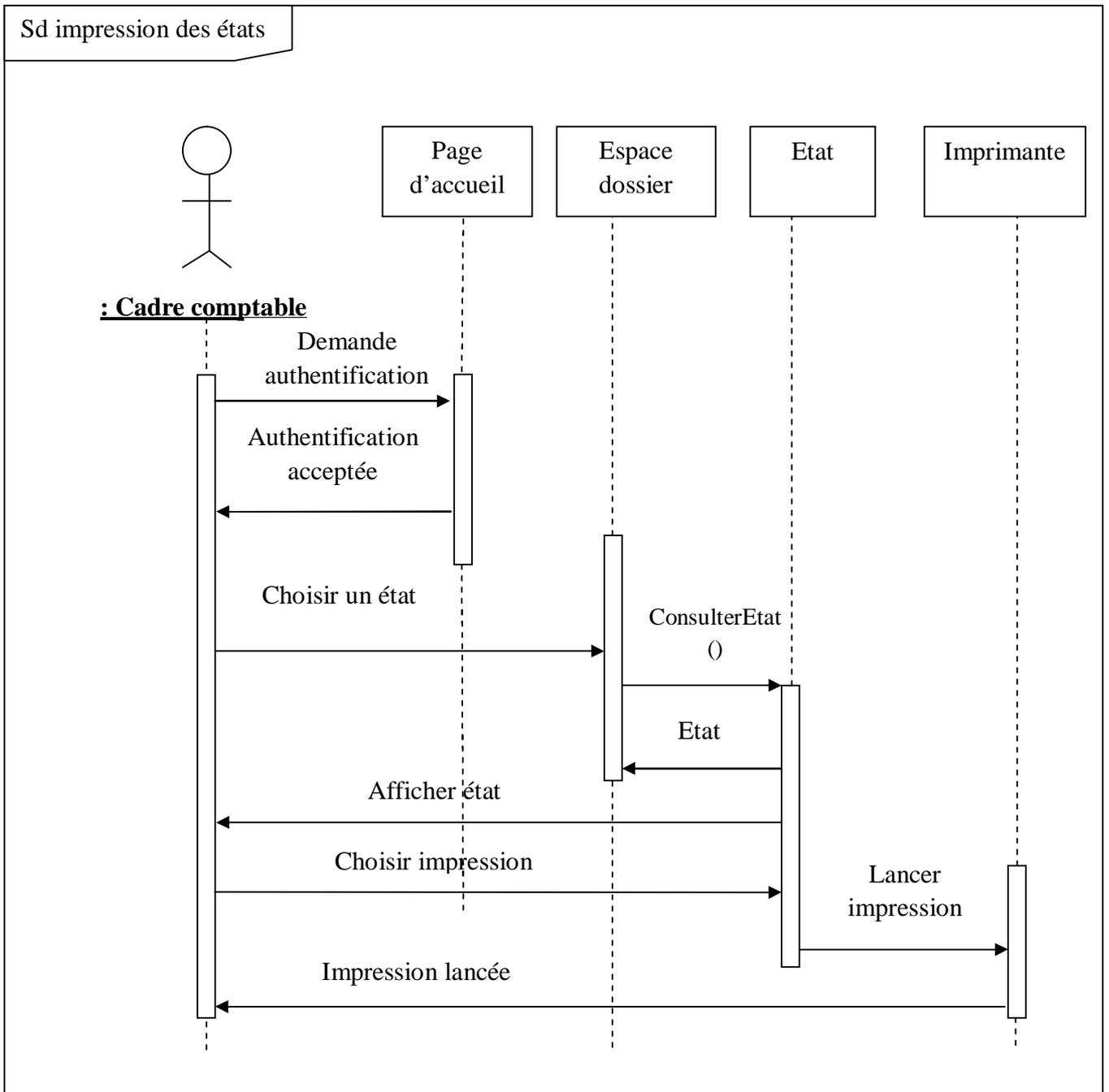


Figure III.17 : Diagramme de séquence du cas d'utilisation « impression d'un état ».

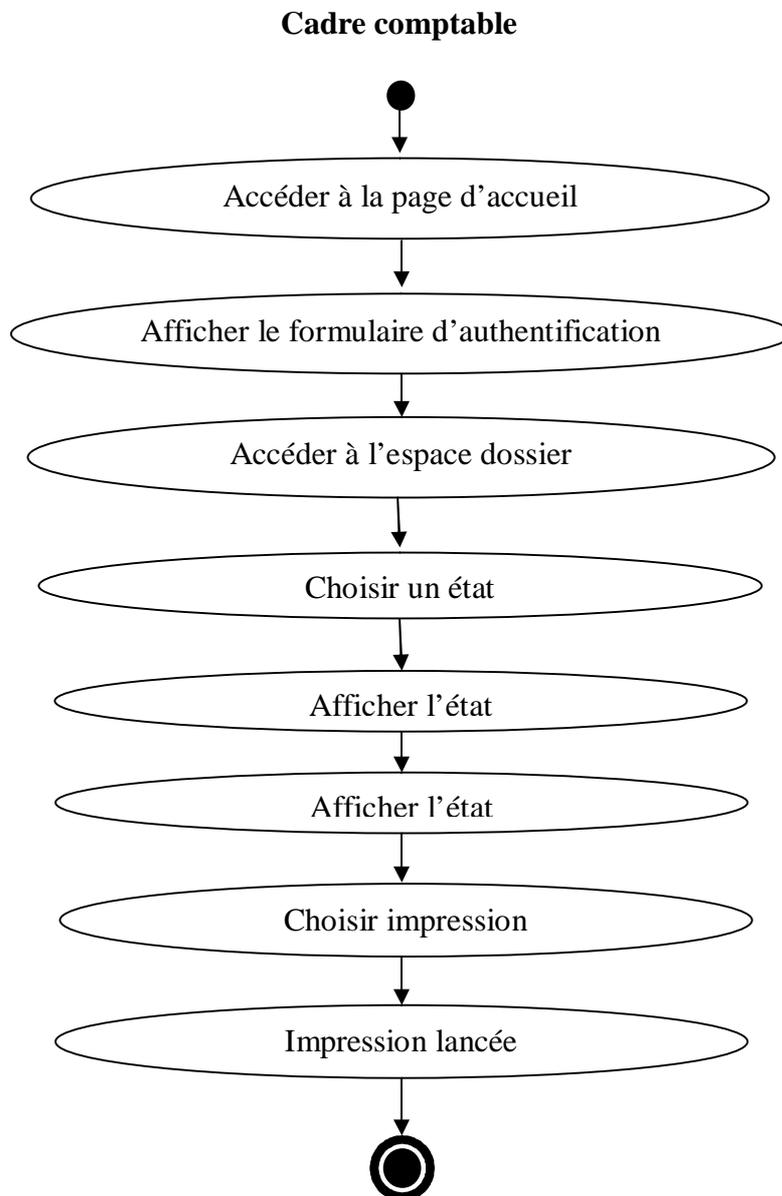


Figure III.18 : Diagramme d'activité du cas d'utilisation « impression d'un état ».

Conclusion :

Ce chapitre a été consacré à l'analyse de l'application avec le langage UML. En premier lieu nous avons présentés les acteurs ainsi que leurs tâches respectives, ensuite nous avons traduits cela sous forme de diagrammes à savoir : diagrammes de cas d'utilisation, de séquences et d'activité.

Introduction :

Dans ce présent chapitre nous allons entamer une partie cruciale du développement logiciel et qui constitue un pont entre la spécification et la réalisation. Elle comporte la conception de l'application ainsi que la conception de la base de données.

I. Architecture de l'application :

L'interaction entre les différentes parties de notre application est présentée par la figure ci-dessous :

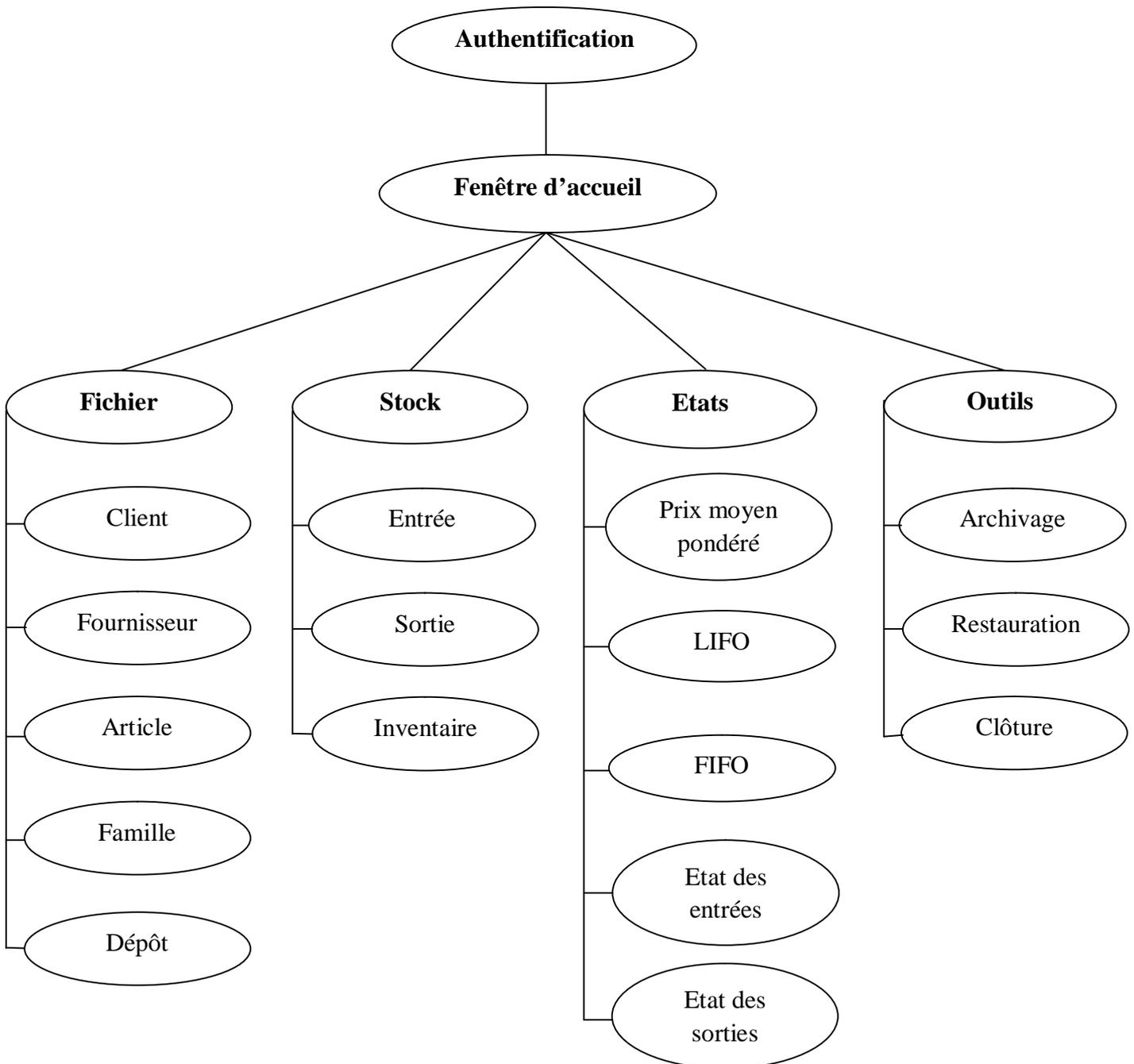


Figure IV.1 : Architecture de l'application

II. Diagramme des classes :

Le diagramme de classes permet de décrire la structure statique du système à l'aide des classes et des relations.

III. Conception de la base de données :

Le modèle relationnel de données est un modèle de données comme d'autres existants ayant pour but de décrire le monde réel à partir des informations et des données qu'on peut en extraire et se différencie par la nature des associations qu'ils permettent de modéliser. L'objectif essentiel du modèle relationnel était d'accroître l'indépendance vis-à-vis du niveau de représentation des données. Du point de vue utilisateur, une base de données peut être considérée comme un ensemble de tables manipulables par des langages de haut niveau dont la caractéristique principale est d'être des langages non procéduraux. Pour ces raisons nous avons choisi d'utiliser une base de données relationnelle.

a. Tableau récapitulatif des classes de la base de données :

A ce niveau de l'application et tout en considérant les besoins déjà spécifiés nous avons envisagé un certain nombre de tables pour enregistrer les données nécessaires pour la gestion de la comptabilité des stocks. Les tables utilisées sont données par le tableau suivant :

Entité	Attributs	Codification	Type	Taille
Client	-Code client	Code_Cli	Varchar	10
	-Raison sociale client	Raison_Cli	varchar	30
	-Adresse client	Adresse_Cli	varchar	50
	-Ville client	Ville_Cli	varchar	15
	-Téléphone client	Tel_Cli	varchar	14
	-Fax client	Fax_Cli	varchar	14
	-Numéro du registre de commerce client	NRC_Cli	varchar	15
	-Article d'imposition client	IA_Cli	varchar	15
	-Matricule fiscal client	MF_Cli	varchar	15
	-Banque client	Banque_Cli	varchar	30
	-Compte client	Cpt_Cli	int	20
	-Adresse banque client	Adresse_bnq_Cli	varchar	30
-Code wilaya client	Code_wil_Cli	int	02	
Fournisseur	-Code fournisseur	Code_Fr	varchar	10
	-Raison social fournisseur	Raison_Fr	varchar	30
	-Adresse fournisseur	Adresse_Fr	varchar	50
	-Ville fournisseur	Ville_Fr	varchar	15
	-Téléphone fournisseur	Tel_Fr	varchar	14
	-Fax fournisseur	Fax_Fr	varchar	14
	-Numéro du registre de commerce fournisseur	NRC_Fr	varchar	15
	-Matricule fiscal fournisseur	MF_Fr	varchar	15
	-Article d'imposition fournisseur	IA_Fr	varchar	15
	-Banque fournisseur	Banque_Fr	varchar	15
	-Adresse banque fournisseur	Adresse_bnq_Fr	varchar	30
	-Numéro du compte fournisseur	N_compte_Fr	int	20
Article	-Référence article	Reference_Art	varchar	20
	-Désignation article	Dsignation_Art	varchar	30
	-Unité de mesure article	Unite_mesure_Art	varchar	10
	-Stock de départ article	Stock_dep_Art	int	10
	-Stock actuel article	Stock_act_Art	int	10
	-Marque article	Mark_Art	varchar	10
	-Modèle article	Modele_Art	varchar	10

	-Prix moyen pondéré -Prix moyen pondéré ancien -Prix de valorisation	PMP_Art PMP_old_Art Prix_val_Art	money money money	
Famille	<u>-Code famille</u> -Nom famille	Code_Fm Nom_Fm	int varchar	05 15
Dépôt	<u>-Code dépôt</u> -Adresse dépôt -Ville dépôt -Téléphone dépôt -Fax dépôt	Code_Dp Adresse_Dp Ville_Dp Tel_Dp Fax_Dp	int varchar varchar varchar varchar	03 30 15 14 14
Dossier	<u>-Code dossier</u> -Libelle dossier	Code_Dos Libelle_Dos	varchar varchar	10 50
Entrée	<u>-Numéro entrée</u> -Date entrée	Num_Ent Date_Ent	int datetime	03 10
Sortie	<u>-Numéro sortie</u> -Date sortie	Num_Srt Date_Srt	int datetime	03 10
Inventaire	<u>-Numéro inventaire</u> -Date inventaire	Num_Inv Date_Inv	varchar datetime	04 10
Mouvement réception	<u>-Code mouvement</u> -Désignation	Code_Mvmt_R Designation_R	varchar varchar	02 40
Mouvement livraison	<u>-Code mouvement</u> -Désignation	Code_Mvmt_L Designation_L	varchar varchar	02 40

Remarque :

Varchar : la donnée est de type chaîne de caractères.

Int : la donnée est un chiffre entier.

Datetime : la donnée est de type date.

Money : La donnée est de type monétaire.

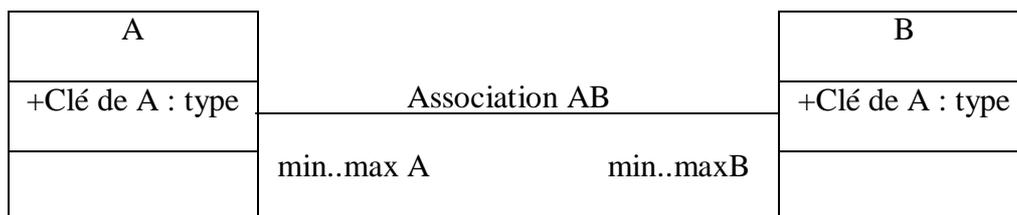
— : un attribut souligné signifie qu'il s'agit d'une clé primaire.

b. Le modèle logique : [B13]

Afin d'assurer la cohérence de la base de données lors du passage du modèle conceptuel (le diagramme de classe) au modèle logique, un certain nombre de règles doivent être respectées. Ces règles sont les suivantes :

- a. **Les classes** : la règle est assez simple, « chaque classe devient une relation ». Chaque attribut de la classe devient un attribut de la relation. Concernant la clé, soit il existe une clé pour la classe qui deviendra de la clé de la relation, ou bien, on ajoute une clé artificielle à la relation.
- b. **Traduire les associations** : l'objectif est de mémoriser les liens entre les objets des classes reliées par l'association. La solution à cette mémorisation dépend de la cardinalité de l'association, plus exactement de ses maximums.

Soient les deux classes A et B suivantes, qui sont reliées par une association AB :



Le tableau suivant examine les cas possibles concernant le maximum des cardinalités de A et de B :

MaxA \ MaxB	1	>1
1	Ajouter la clé de B dans la relation de A comme attribut ou Ajouter la clé de A dans la relation de B comme attribut.	Ajouter la clé de B dans la relation de A comme attribut.
>1	Ajouter la clé de A dans la relation de B comme attribut.	Créer une relation AB ayant comme attribut la clé de A et la clé de B.

En se basant sur les règles citées précédemment pour le passage du modèle conceptuel au modèle logique, nous obtenons les relations suivantes :

Client (Code_Cli, Raison_Cli, Adresse_Cli, Ville_Cli, Tel_Cli, Fax_Cli, NRC_Cli, IA_Cli, MF_Cli, Banque_Cli, Cpt_Cli, Adresse_bnq_Cli, Code_wil_Cli).

Fournisseur (Code_Fr, Raison_Fr, Adresse_Fr, Ville_Fr, Tel_Fr, Fax_Fr, NRC_Fr, MF_Fr, IA_Fr, Banque_Fr, Adresse_bnq_Fr, N_compte_Fr).

Article (Reference_Art, Dsignation_Art, Unite_mesure_Art, Stock_dep_Art, Stock_act_Art, Mark_Art, Modele_Art, PMP_Art, PMP_old_Art, Prix_val_Art, Code_Fm*).

Famille (Code_Fm, Nom_Fm).

Dépôt (Code_Dp, Adresse_Dp, Ville_Dp, Tel_Dp, Fax_Dp, Code_Dos*).

Dossier (Code_Dos, libelle_Dos).

Entrée (Num_Ent, Date_Ent, Code_Fr*, Code_Mvmt_R*, Code_Dos*).

Sortie (Num_Srt, Date_Srt, Code_Cli*, Code_Mvmt_L*, Code_Dos*).

Inventaire (Num_Inv, Date_Inv).

Mouvement Réception (Code Mvmt R, Designation _R).

Mouvement Livraison (Code Mvmt L, Designation _L).

Article Dépôt (Code Dp, Reference Art, Quantité Dp).

Article Inventaire(Num Inv, Reference Art, Code Dp, Quantite Inventoriee).

Article Sortie(Num Srt, Reference Art, Code Dp, Quantite).

Article Entrée(Num Ent, Reference Art, Code Dp, Quantite).

Remarque :

— : un attribut souligné signifie qu'il s'agit d'une clé primaire.

* : un attribut suivi d'une étoile signifie qu'il s'agit d'une clé étrangère.

Conclusion :

Etant donné les besoins des utilisateurs de notre application, la phase de conception vient pour permettre la détermination des différents objets contribuant à assurer les fonctionnalités souhaitées. Cette phase est une préparation à la phase de réalisation garantissant une organisation claire et précise ainsi qu'une facilité d'implémentation des classes invoquées, des structures de données utilisées et les relations qui existent entre les différentes classes. Nous essayons dans le chapitre réalisation d'implémenter les différentes classes et de montrer les fonctionnalités réalisées suite à cette implémentation.

Introduction :

Dans cette partie de notre étude, nous allons concevoir une application qui permettra de répondre aux besoins précédemment identifiés. Cette application gèrera de manière automatique la gestion de la comptabilité des stocks des produits finis.

Nous avons choisi de travailler avec l'environnement de développement intégré Delphi 7. Dans ce qui suivra, nous allons présenter l'environnement technique de développement.

I. Description du matériel utilisé :

Pour la réalisation de notre application, nous avons utilisé un micro ordinateur qui présente les caractéristiques suivantes :

- **micro processeur** : Pentium (R) Dual-Core CPU T4300
- **RAM** : 3 GO
- **Fréquence d'horloge** de 2.2GHZ
- **Disque dur** de 320GO
- **Clavier** Azerty
- **Ecran** HD Brightview
- **système d'exploitation** : Windows 7

II. Présentation du langage de programmation utilisé : [B14]

La programmation orientée objet relève d'une conception ascendante définie comme des « messages » échangés par des entités de base appelées : objets.

Les langages objets sont fondés sur la connaissance d'une seule catégorie d'entité informatique : l'objet. Dans un objet, traditionnellement ce sont les données qui deviennent prépondérantes. On se pose d'abord la question : « de quoi parle-t-on ? » et non pas la question « que veut-on faire ? », comme en programmation structurée modulaire peuvent être considérées comme des pré-objets Delphi 7 est un RAD visuel (Rapid Application Development) fondé sur une extension orientée objet, visuelle et événementielle de Pascal. Il est considéré comme un prolongement intéressant de ce langage.

1. Notion de programmation visuelle et événementielle :**• Programmation visuelle :**

Actuellement dans le développement d'un logiciel, un temps très important est consacré à l'ergonomie et la communication, cette part ne pourra que grandir dans un avenir proche ; car les utilisateurs veulent s'adresser à des logiciels efficaces mais aussi conviviaux et faciles d'accès.

A ce jour la programmation visuelle est l'une des réponses à cette attente des développeurs. Un langage de programmation visuelle permet « d'écrire » la partie communication d'un programme uniquement avec des dessins, diagramme, icônes... etc.

Autrement dit, un langage de base qui, derrière chaque action visuelle (dépôt de contrôle, click de souris, modification des propriétés,...etc) engendre des lignes de codes automatiquement et d'une manière transparente au développeur.

• Programmation événementielle :

La programmation orientée événements est une logique de conception selon laquelle un programme est construit avec des objets et leurs propriétés et d'après laquelle les interventions de l'utilisateur sur les objets du programme déclenchent l'exécution des routines associées.

Il existe deux approches pour construire un programme :

a. L'approche événementielle :

Qui intervient principalement dans l'interface entre le logiciel et l'utilisateur, mais aussi dans la liaison dynamique du logiciel avec le système. Et enfin dans la sécurité.

b. L'approche visuelle :

Nous aide et simplifie notre tâche dans la construction du dialogue homme machine. La combinaison de ces deux approches produit un logiciel habillé et adapté au système d'exploitation.

2. Interface de développement « Delphi 7 » :

En Delphi 7 pour créer une interface homme-machine, il faut utiliser des fiches et des contrôles. Ces fiches sont des objets au sens informatique de la programmation orientée objet, mais elles possèdent une représentation visuelle.

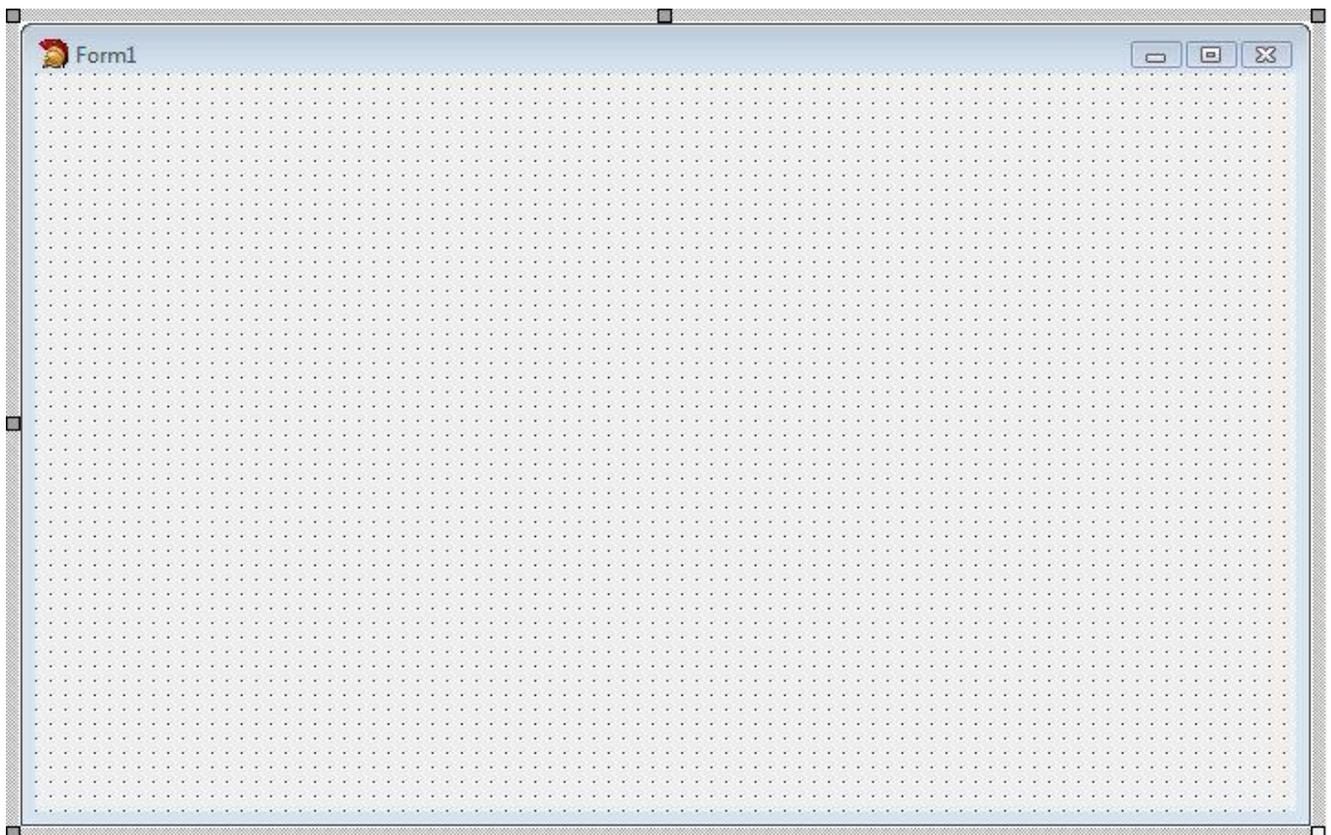
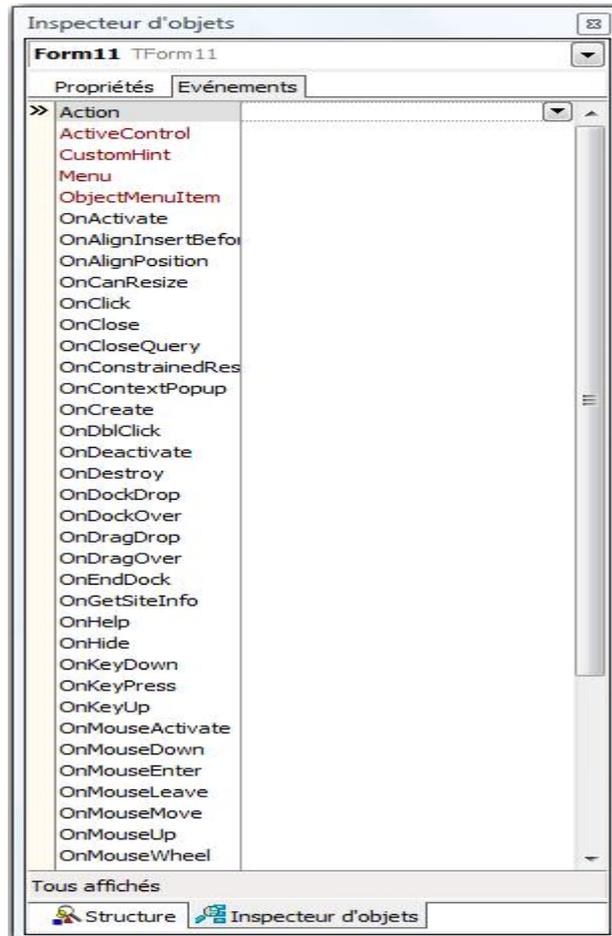


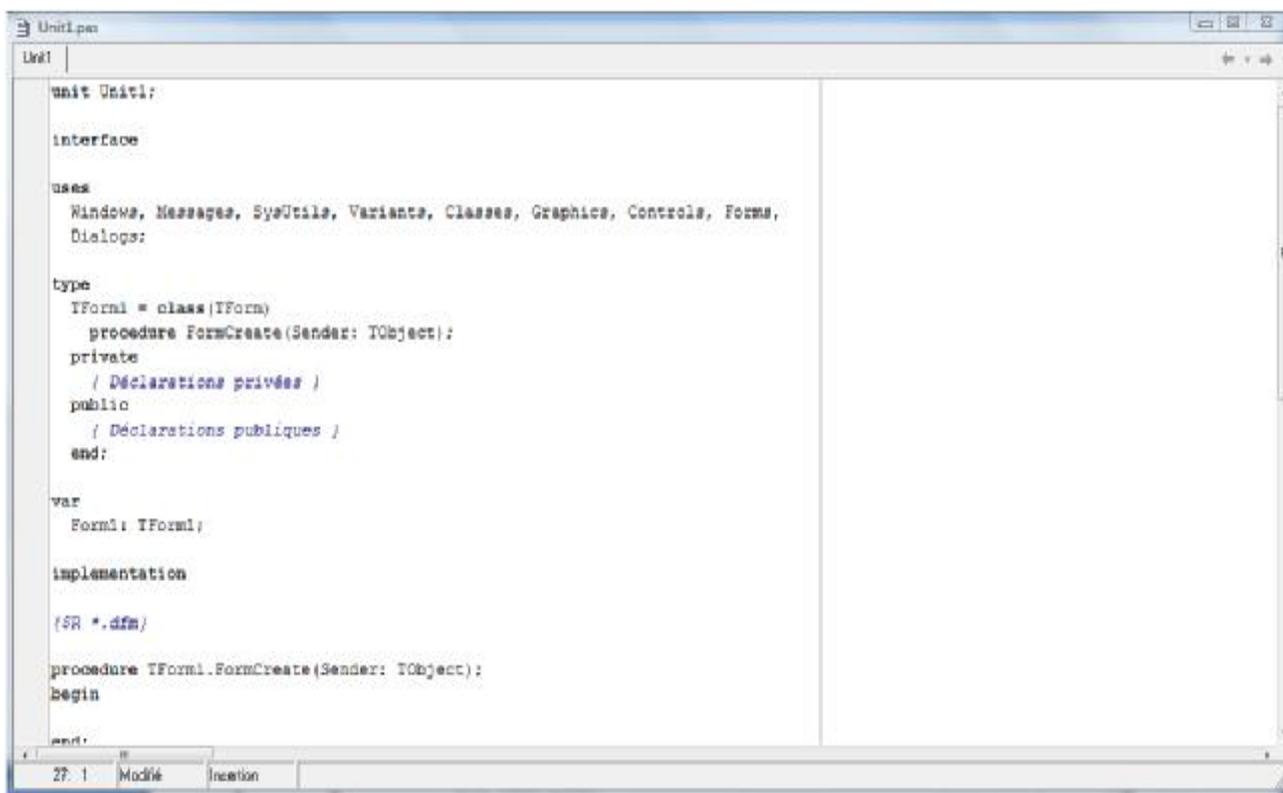
Figure V.1 : Interface de développement « Delphi7 »

La description de chaque fiche est stockée dans un fichier fiche. Ces fichiers la décrivent chaque composante de la fiche, y compris les valeurs de toutes les propriétés persistantes. On ne spécifie pas le fichier fiche par programme, on crée tout simplement la fiche en sélectionnant des

composantes dans la palette de composantes et on les personnalisant dans l'inspecteur d'objets pour qu'ils répondent aux besoins par des propriétés et des événements.



Chaque fiche d'un objet est également associée à une unité. L'unité contient le code source des gestionnaires attachés aux événements de la fiche et des composants qu'elle contient.



3. Delphi et les bases de données :

Une base de données est composée de données stockées dans des mémoires de masse sous une forme structurée, et accessibles par des applications différentes et des utilisateurs différents. Une base de données doit pouvoir être utilisée par plusieurs utilisateurs en même temps.

Delphi 7 intègre toute une collection d'outils destinés à la création et la gestion de base de données. Cependant, nous avons fait le choix de travailler avec SQL server 2005 qui est une très bonne manière d'interroger la base de données, avec des instructions simples, il offre des possibilités d'utilisation avec des bases de données volumineuses.

4. SQL Server 2005 :

SQL Server 2005 (Nom de code : Yukon) est l'actuel SGBDR (Système de Gestion de Base de Données Relationnelles) de la plateforme Microsoft.

Basé sur les points forts de son prédécesseur (SQL Server 2000), Yukon inclus beaucoup de nouvelles fonctionnalités qui permettent aux entreprises de devenir plus productives. Il permet, entre autres choses :

- § De créer et déployer des applications plus sûres, plus puissantes et plus fiables ;
- § De proposer aux développeurs un environnement de développement riche, souple et moderne permettant de créer des applications de bases de données plus sûres ;
- § De partager des données entre diverses plates-formes, applications et systèmes pour faciliter les connexions, tant internes qu'externes.

SQL Server 2005 a apporté des changements significatifs dans les secteurs suivants :

- § La montée en charge : des améliorations comme le partitionnement, l'isolement des captures instantanées et la prise en charge des systèmes 64 bits, permettent de créer et de déployer les applications les plus exigeantes.
- § Sécurité : cryptage de base de données, choix de la sécurité maximale par défaut.
- § Gestion : une nouvelle suite d'outils de gestion de base de données, amélioration du profiler SQL.
- § Interopérabilité : grâce à une prise en charge étendue des standards, des services Web et de Microsoft.Net Framework, SQL Server 2005 assure l'interopérabilité entre plates-formes, application et systèmes.

D'un point de vue de développement, Yukon intègre également les nouveautés :

- § Amélioration des outils : les développeurs pourront utiliser un outil de développement pour Transact-SQL, XML, MDX (Multidimensional Expression) et XML for Analysis(XML/A). L'intégration avec l'outil de développement Visual Studio permettra un développement et un débogage plus efficace des applications métier et décisionnelles.
- § Intégration de la CLR (Common Language Runtime) dans le moteur de base de données : les développeurs auront la possibilité de choisir le langage (C#, VB.NET, Transact-SQL) qu'ils utiliseront pour développer leurs application de base de données.
- § Support du XML et XQuery ;
- § Support des Web Services : la prise en charge de standards ouverts, existant ou nouveaux, tels que http (Hypertext Transfer Protocol), XML, SOAP (Simple Object Access Protocol), XQuery et XSD (XML Schema Definition) facilitera également les communications entre systèmes d'entreprise étendus.

III. Interfaces logiciel :

§ Authentification des utilisateurs :

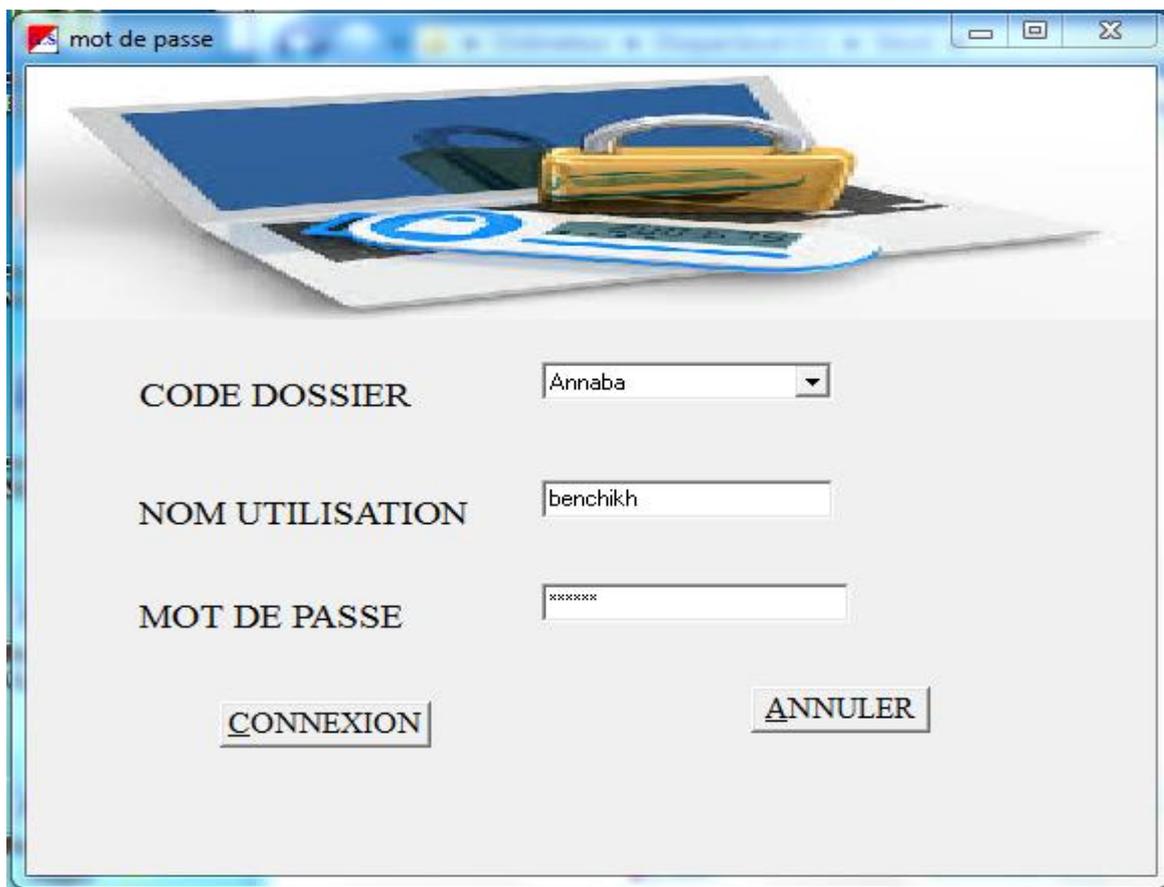


Figure V.2 : Interface d’authentification

§ Espace dossiers :

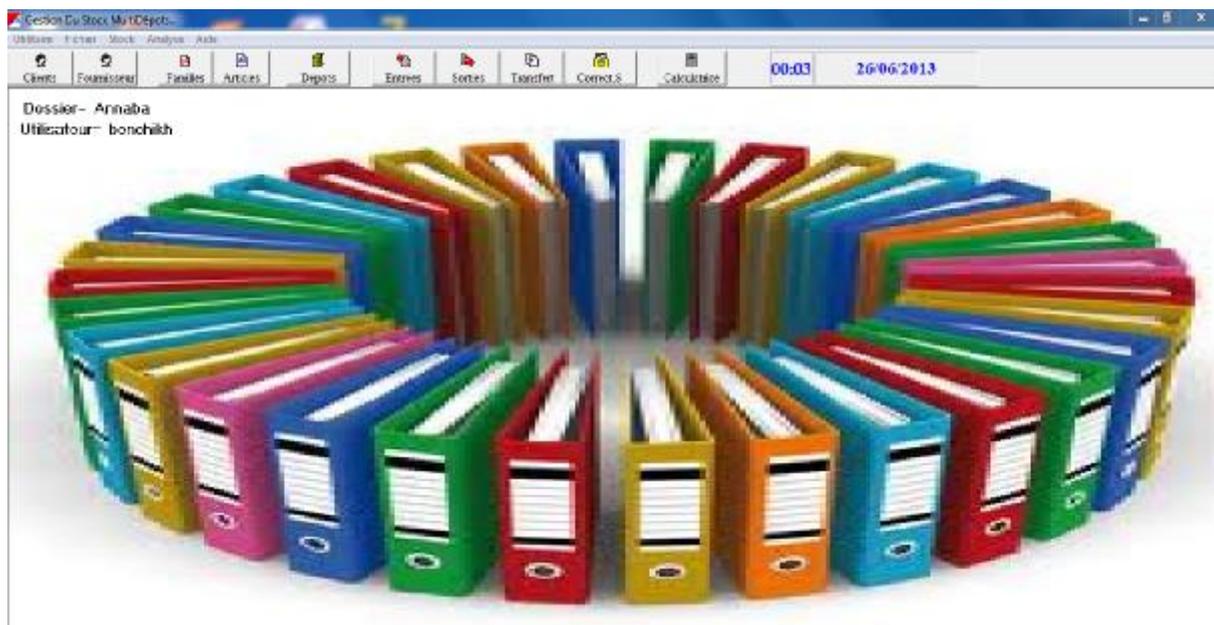


Figure V.3 : Interface d’accueil

§ ajouter un article :

The 'Article' dialog box contains the following fields:

- Référence Article** (Grouped box):
 - Référence :
 - Désignation :
 - Famille :
 - Marque_article :
 - Référenceec_art :
- Unité de mesure : Poids :
- Prix d'Achat : coefficient de marge :
- Prix de Vent :
- Stock_depart : Stock_actuel :
- Pmp_old : Pmp :
- Prix_val : Tva :
- Numero compte :

Buttons:

Figure V.4 : Interface d'ajout d'un article

§ Ajouter une famille :

The 'Saisir Une Familles' dialog box contains the following fields:

- Code :
- Famille :

Buttons:

Figure V.5 : Interface d'ajout d'une famille

§ **Ajouter un fournisseur :**

Fournisseurs ...

CODE:

Raison Social :

Adresse :

Ville :

Tél : Fax :

Référence Commercial | **Référence Bancaire**

Numero Registre Commercial :

Matricule Fiscal :

Article Imposition :

Figure V.6 : Interface d'ajout d'un fournisseur

§ **Ajouter un dépôt :**

Dépôts du Stocke...

Code :

Libelle :

Adresse :

Ville :

Figure V.7 : Interface d'ajout d'un dépôt

§ Ajouter un client :

Client ...

CODE:
 Raison Social:

Adresse:
 Ville:

Tél: Fax:

Référence Commercial | **Référence Bancaire** | **Parametre**

Numero Registre Commercial:
 Matricule Fiscal:
 Article Imposition:
 Créance_initiale:
 Créance_restante:

Figure V.8 : Interface d’ajout d’un client

§ Ajouter une entrée :

Entrées

N° Entree: Date: Mouvement: code: Total HT:

Fournisseur: Code:

Référence	Désignation	Cuts_Depot	Non_depot	Prix d'Achat	Quantité	Total HT

Figure V.9 : Interface d’ajout d’une entrée

§ Ajouter une sortie :

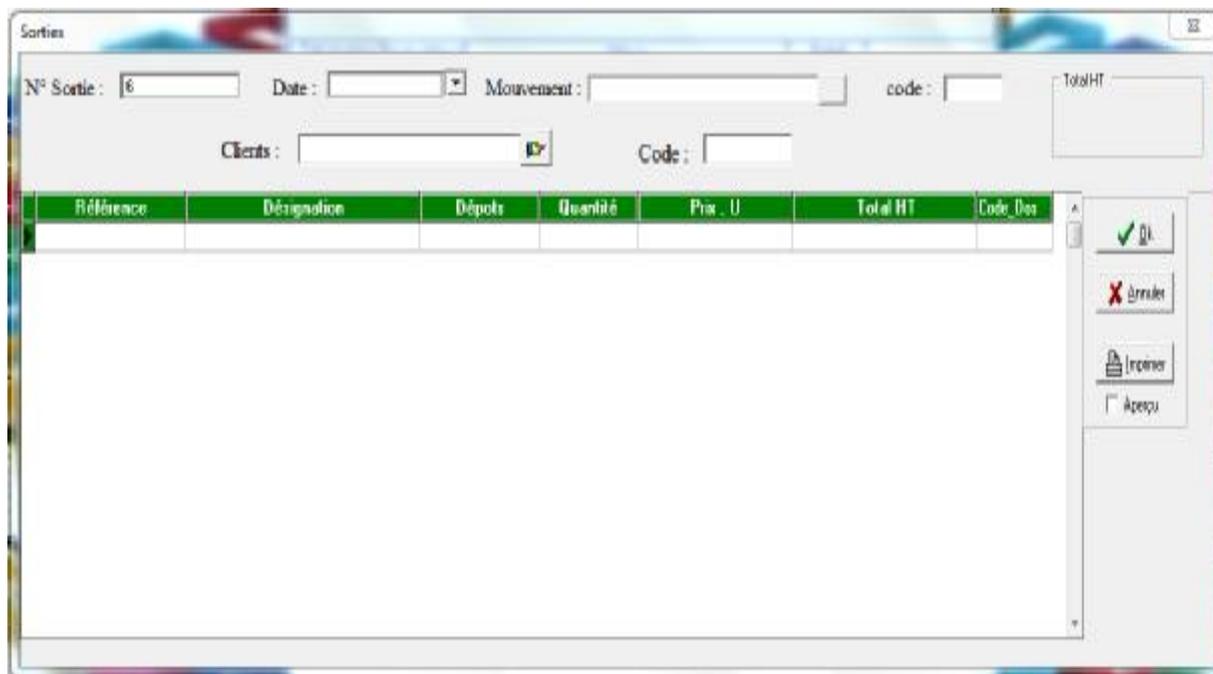


Figure V.10: Interface d'ajout d'une sortie

§ Interface de recherche d'un mouvement de sortie :



Figure V.11: Interface de recherche d'un mouvement de sortie

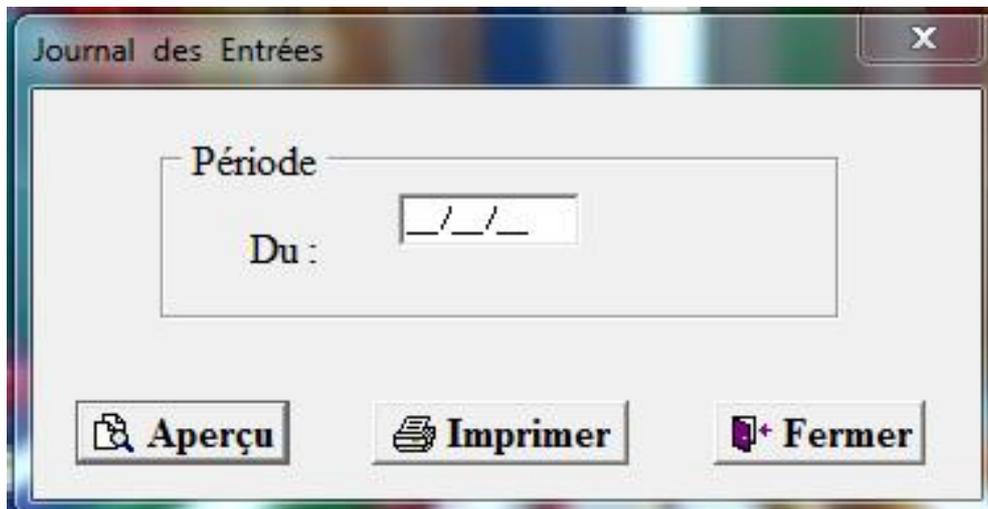
§ Interface de journal des entrées :

Figure V.12: Interface de journal des entrées

Conclusion :

Dans ce chapitre, nous avons en premier lieu présenté l'environnement et les outils utilisés pour implémenter notre application. Ensuite nous avons donné quelques exemples d'interfaces pour illustrer les principales fonctionnalités de notre application.

CONCLUSION GÉNÉRALE ET PERSPECTIVES:

La réalisation de ce projet a permis de mettre en place une application client/serveur multi-dossiers de comptabilisation des stocks des produits finis. Notre rôle était donc d'implémenter tous les modules qui constituent notre application.

La réalisation de ce travail nous a permis d'élargir aussi bien nos connaissances théoriques que pratiques en rapport avec les applications de gestion, et d'acquérir de nouvelles connaissances sur le langage de programmation « Delphi 7 », le langage de modélisation « UML » et d'améliorer nos connaissances sur la gestion des bases de données.

En instaurant la politique de la confiance mutuelle, chacun apportant sa contribution au projet grâce à la communication pour la réussite de notre projet. Nous avons pu réussir la tâche qui nous a été confiée. Néanmoins notre application peut être améliorée et être étendue par l'intégration d'autres outils.

Bibliographie

[B01] : Manuel Entreprise Nationale des Industries de l'Electroménager Département Finance et Comptabilité (Unité Commerciale). Année 2008.

[B02] : Module de gestion des stocks, formation ANSEJ.

[B03] : Casimir NTEZIRYIMANA, L'analyse du système de gestion des stocks dans une organisation. Cas d'ACTED (Agence d'aide à la coopération et au développement), Institut de gestion et de développement économique.

[B06] : B. AUGÉ et A.VERNHET. Le système d'information de la comptabilité de gestion.

[B07] : J.GABAY et D.GABAY. UML 2 analyse et conception. Edition Dunod, Paris, 2008.

[B09] : B. PETIT, Architecture des réseaux, édition 2008.

[B10] : Mémoire (Ingénieur) réalisé par M^{elle} Benchabane Bahia et M^{elle} Belhadj Samia

Thème : conception et réalisation d'un portail web collaboratif. Cas : la direction coordination group France de Sonatrach (DG) 2008/2009

[B11] : polycopié complément de cours communication de données et réseaux M^r JEMAH 4^{ième} année ingénieur promotion 2008/2009

[B12] : fiche d'emploi ENIEM.

[B13] : Mémoire (Master2) réalisé par M^{er} Belkacemi Anis

Thème : conception et réalisation d'une plateforme pour le travail collaboratif. Cas : SADAREP TIZI-OUZOU. Promotion 2011/2013

[B14] : R. Michel DISCALA. Programmation orientée objet : DELPHI, volume1, B. Edition.

Webographie :

[w04] : <http://www.lacompta.org/vocabulaire/journale.php>

[w05] : <http://www.thesaurus.gouv.qc.ca/tag/terme.do?id=6185>

[w08] : <http://www.apprendre-informatique.com/forum/toile/reseau/typr-reseau/>

Présentation de la méthode UML : [B07]

Introduction :

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

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

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

I. Définition d'UML :

UML (Unified Method Language) a été créé en fusionnant trois méthodes d'analyses et de conception objet à savoir : la méthode OMT (Objet Modeling Technique) de James Rumbaugh, Booch de Grady Booch et OOSE de Ivar Jacobson.

UML est donc une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG (Object Management Group), instance de normalisation internationale du domaine de l'objet.

II. Règles générales :

Afin d'assurer un bon niveau de cohérence et d'homogénéité sur l'ensemble des modèles.

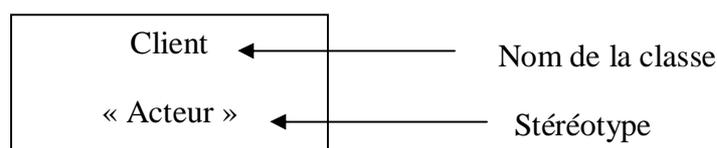
UML propose d'un côté un certain nombre de règles d'écriture ou de représentation graphiques normalisées et d'un autre côté des mécanismes ou des concepts communs applicables à l'ensemble des diagrammes.

Les principaux éléments d'UML que nous présentons sont : le stéréotype, la valeur marquée, la note, la contrainte.

Ø **Le stéréotype** : est prévu pour assurer une réelle capacité d'adaptation et d'évolution de la notation notamment pour prendre en compte les particularités des différentes situations à modéliser.

Par exemple, dans le diagramme de classe, le stéréotype permet de considérer de nouveaux types de classes. Cette possibilité d'extension pour les classes se définit au niveau méta-classe.

Formalisme :



- Ø **La valeur marquée** : UML permet d'indiquer des valeurs particulières au niveau des éléments de la modélisation et en particulier pour les attributs de classe.

Une valeur marquée se définit dans méta-attribut.

Formalisme :

La valeur marquée est mise entre { } avec indication du nom et de la valeur.

Exemple : {persistence : string}.

- Ø **La note** : une note correspond à un commentaire.

Formalisme :



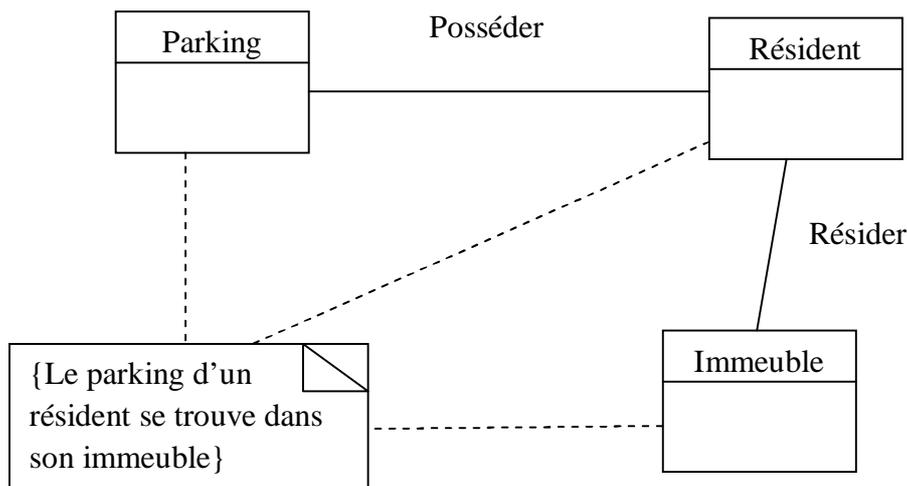
- Ø **La contrainte** : c'est une note ayant une valeur pour un élément de modélisation.

Dans le cas où la contrainte concerne une seule classe, elle s'écrit entre { }, par contre, dans le cas où elle concerne deux classes ou plus, elle s'écrit dans une note.

Formalisme :

Cas 1 : {contrainte}

Cas 2 :



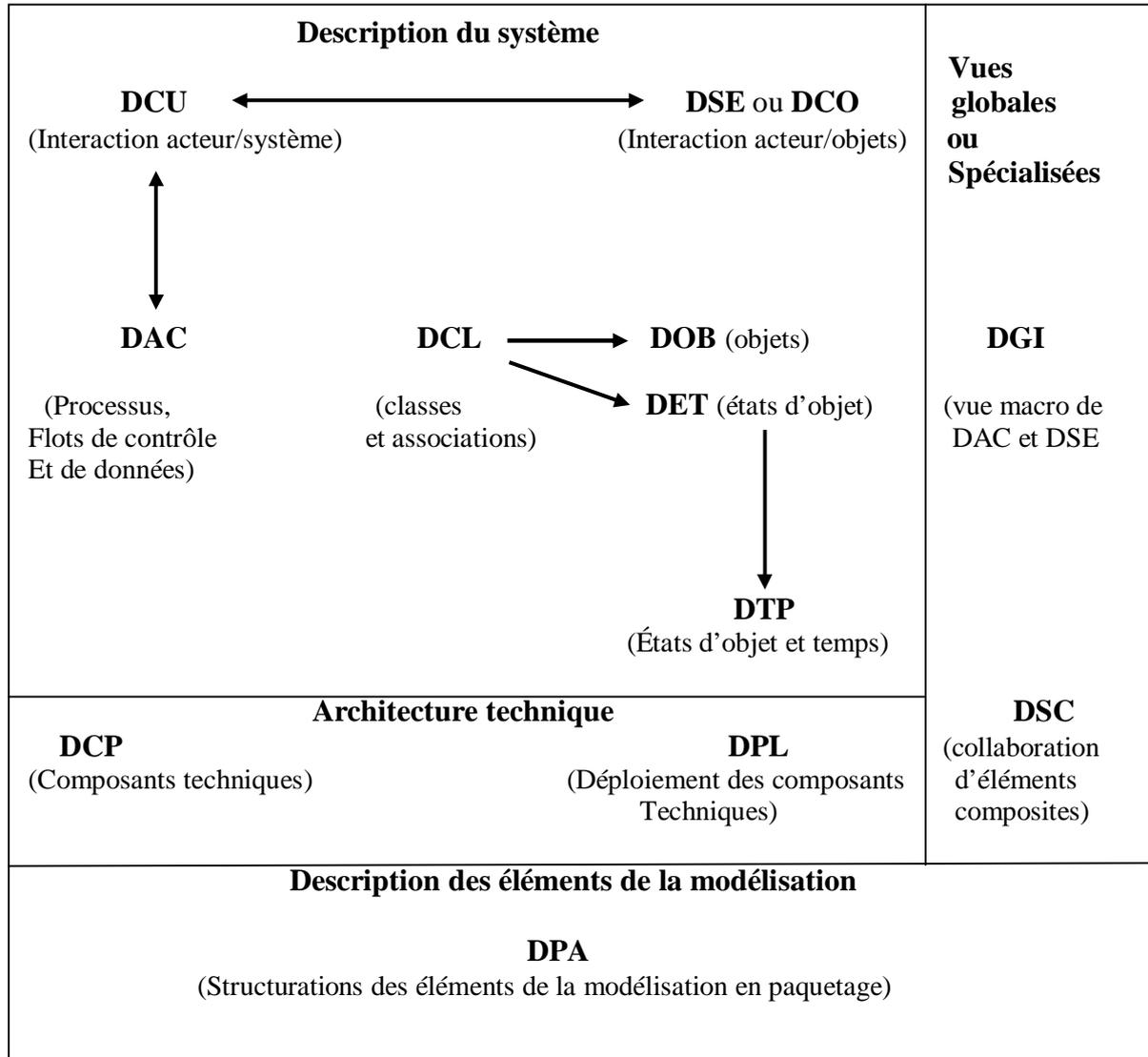
III. Schéma d'ensemble des treize diagrammes d'UML :

Afin de donner quelques points de repères sur le positionnement et les liens entre tous les diagrammes d'UML, nous donnons ici un regroupement des diagrammes en quatre ensembles suivant leur finalité :

- Description du système : huit diagrammes ;
- Architecture technique : deux diagrammes ;

- Vue globale ou spécialisées : deux diagrammes ;
- Partition d'éléments de la modélisation : un diagramme.

Le schéma suivant représente les treize diagrammes en les répartissant sur les quatre ensembles définis :



Les abréviations adoptées pour ce schéma sont définies comme suit :

DAC : diagramme d'activité.

DCL : diagramme de classe.

DOB : diagramme d'objet.

DCP : diagramme de composant.

DCU : diagramme des cas d'utilisation.

DCO : diagramme de communication.

DET : diagramme d'état-transition.

DGI : diagramme globale d'interaction.

DPA : diagramme de paquetage.

DPL : diagramme de déploiement.

DSC : diagramme de structure composite.

DES : diagramme de séquence.

DTP : diagramme de temps.

IV. Présentation des diagrammes utilisés dans le projet :

IV.1. Diagramme des classes :

IV.1.1. Définition : le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe.

IV.1.2. Description du diagramme de classe :

La description du diagramme de classe est fondée sur :

- Le concept d'objet,
- Le concept de classe comprenant les attributs et les opérations,
- Les différents types d'association entre classes.

1. Classe :

Une classe décrit un groupe d'objet ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).

Un objet est une instance d'une classe. La classe représente l'abstraction de ses objets. Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond une adresse mémoire.

Formalisme graphique d'une classe :

Nom de la classe
Attributs
Opérations
Responsabilité et/ou exception

2. **Attribut** : un attribut est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur (sauf cas d'attribut multivalués).

Formalisme :

Nom de la classe
Nom et caractéristique attribut 1
Nom et caractéristique attribut 2
....

▼ **Caractéristiques d'un attribut** :

Le nom de la classe peut être qualifié par un « stéréotype ». la description complète des attributs d'une classe comporte un certain nombre de caractéristiques qui doivent respecter le formalisme suivant :

- **Visibilité/ Nom attribut : type [= valeur initiale {propriétés}]**
 - Visibilité : elle est définie plus loin dans notre exposé.
 - Nom de l'attribut : nom unique dans la classe.
 - Type : type primitif (entier, chaîne de caractère...) dépendent des types disponibles dans le langage d'implémentation ou type classe matérialisant un lien avec une autre classe.
 - Valeur initiale : valeur facultative donnée à l'initialisation d'un objet de la classe.
 - {propriétés} : valeurs marquées facultatives.

Un attribut peut avoir des valeurs multiples. Dans ce cas, cette caractéristique est indiquée après le nom de l'attribut.

Un attribut dont la valeur peut être calculée à partir d'autres attributs de la classe est un attribut dérivé qui se note « /nom de l'attribut dérivé ».

3. **Opération** : une opération est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet. Une méthode est l'implémentation d'une opération.

Formalisme :

Nom de la classe
Nom et caractéristique attributs
....
Nom et caractéristique opération 1
Nom et caractéristique opération 2
....

▼ Caractéristiques d'une opération :

La description complète des opérations d'une classe comporte un certain nombre de caractéristique qui doivent respecter le formalisme suivant :

- **Visibilité Nom d'opération (paramètres) [: [type résultat] {propriétés}]**
 - Visibilité : elle est défini plus loin dans notre exposé.
 - Nom d'opération : utiliser un verbe représentant l'action à réaliser.
 - Paramètres : liste de paramètres (chaque paramètre peut être décrit, en plus de son nom, par son type et sa valeur par défaut). L'absence de paramètre est indiquée par ().
 - Type résultat : type de (s) valeur (s) retourné (s) dépendant des types disponibles dans le langage d'implémentation. Par défaut, une opération ne retourne pas de valeur, ceci est indiqué par exemple par le mot réservé «void» dans le langage C++ ou Java.
 - {propriétés} : valeurs facultatives applicable.

4. Visibilité des attributs et des opérations :

Chaque attribut ou opération d'une classe peut être de type public, protégé, privé ou paquetage. Les symboles + (public), # (protégé), - (privé) et ~ (paquetage) sont indiqués devant chaque attribut ou opération pour signifier le type de visibilité autorisé pour les autres classes.

Les droits associés à chaque niveau de confidentialité sont :

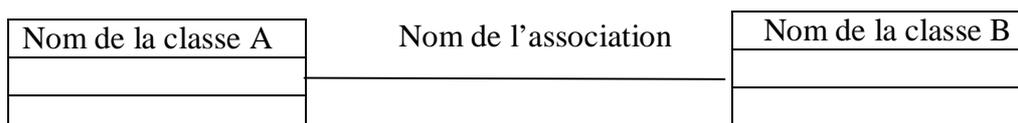
- **Public (+)** : attribut ou opération visible par tous.
- **Protégé (#)** : attribut ou opération visible seulement à l'intérieur de la classe et pour toute les sous-classes de la classe.
- **Privé (-)** : attribut ou opération seulement visible à l'intérieur de la classe.
- **Paquetage (~)** : attribut ou opération ou classe seulement visible à l'intérieur du paquetage où se trouve la classe.

5. Association :

Un lien est une connexion physique ou conceptuelle entre instances de classes donc entre objets. Une association décrit un groupe de liens ayant une même structure et une même sémantique. Un lien est une instance d'une association. Chaque association peut être identifiée par son nom.

Une association entre classes représente les liens qui existent entre les instances de ces classes.

Formalisme :

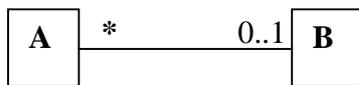


6. Multiplicité :

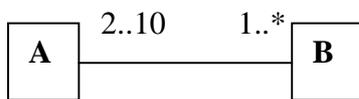
La multiplicité indique un domaine de valeurs pour préciser le nombre d'instances d'une classe vis-à-vis d'une autre classe pour une association donnée. La multiplicité peut être utilisée pour d'autres usages comme par exemple un attribut multivalué. Le domaine de valeurs est décrit selon plusieurs formes.

- **Intervalle fermé** : exemple : 2,3..15.
- **Valeurs exacte** : exemple : 3, 5,8.
- **Valeur indéterminé notée *** : exemple : 1..*.
- Dans le cas où l'on utilise seulement *, cela traduit une multiplicité 0..*.
- Dans le cas de multiplicité d'association, il faut indiquer les valeurs minimales et maximales d'instances d'une classe vis-à-vis d'une instance d'une autre classe.

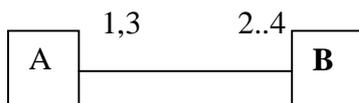
Formalisme :



- A une instance de A correspond 0 ou 1 instance de B.
- A une instance de B correspond 0 à un nombre non déterminé d'instances de A.



- A une instance de A correspond 1 à plusieurs nombre non déterminé d'instances de B.
- A une instance de B correspond 2 à 10 instances de A.

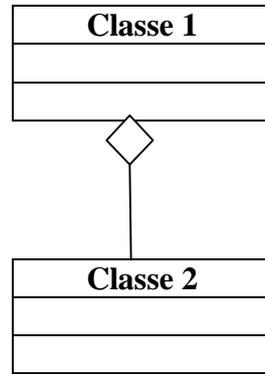


- A une instance de A correspond 2 à 4 instances de B.
- A une instance de B correspond 1 ou 3 instances de A.

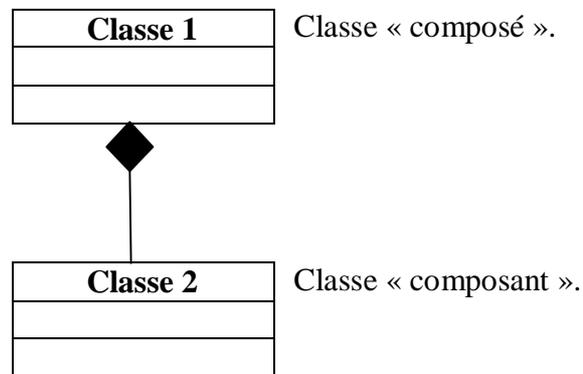
IV.1.3. Agrégation et composition entre classes :

a. Agrégation :

L'agrégation est une association qui permet de représenter un lien de type « ensemble » comprenant des « éléments ». Il s'agit d'une relation entre une classe représentant le niveau « ensemble » et 1 à n classes de niveau « éléments ». L'agrégation représente un lien structurel entre une classe et une ou plusieurs autres classes.

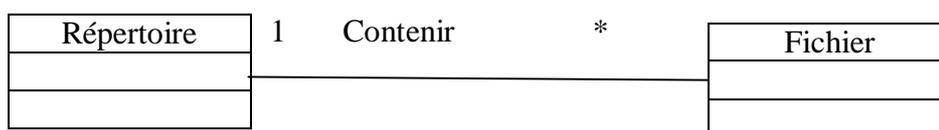
Formalisme :**b. Composition :**

La composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre la classe « composant » et la ou les classes « composé ». Autrement dit la suppression de la classe « composé » implique la suppression de la ou des classes « composant ».

Formalisme :**IV.1.4. Association qualifiée, dépendance et classe d'interface :****a. Qualification :**

La qualification d'une relation entre deux classes permet de préciser la sémantique de l'association et de qualifier de manière restrictive les liens entre les instances.

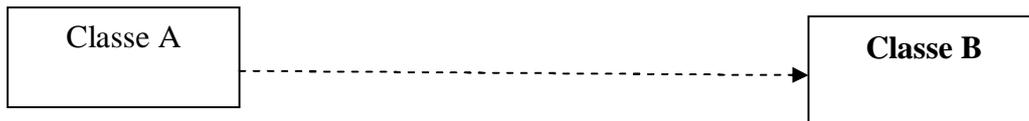
Seules les instances possédant l'attribut indiqué dans la qualification sont concernées par l'association. Cet attribut ne fait pas parti de l'association.

Formalisme :

b. Dépendance :

La dépendance entre deux classes permet de représenter l'existence d'un lien sémantique. Une classe B est en dépendance de la classe A si des éléments de la classe A sont nécessaire pour construire la classe B.

Formalisme :



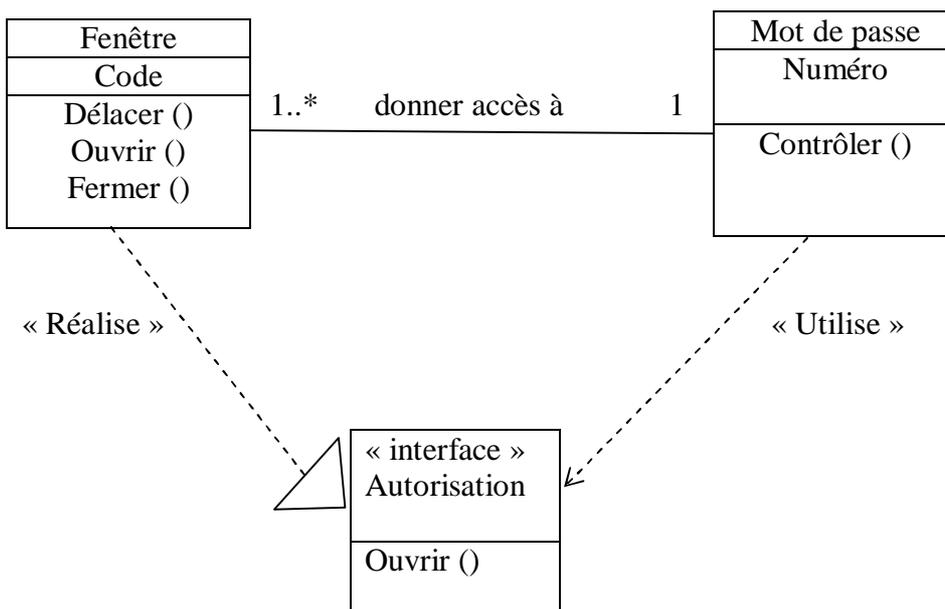
c. Interface :

Une classe d'interface permet de décrire la vue externe d'une classe. La classe d'interface, identifiée par un nom, comporte la liste des opérations accessible par les autres classes. Le compartiment des attributs ne fait pas parti de la description d'une interface.

L'interface peut être aussi matérialisée plus globalement par un petit cercle associé à la classe source.

La classe utilisatrice de l'interface est reliée au symbole de l'interface par une flèche en pointillé. La classe d'interface est une spécification et non une classe réelle. Une classe d'interface peut s'assimiler à une classe abstraite.

Formalisme :



IV.1.5. Généralisation et spécialisation :

La généralisation/spécialisation et l'héritage simple :

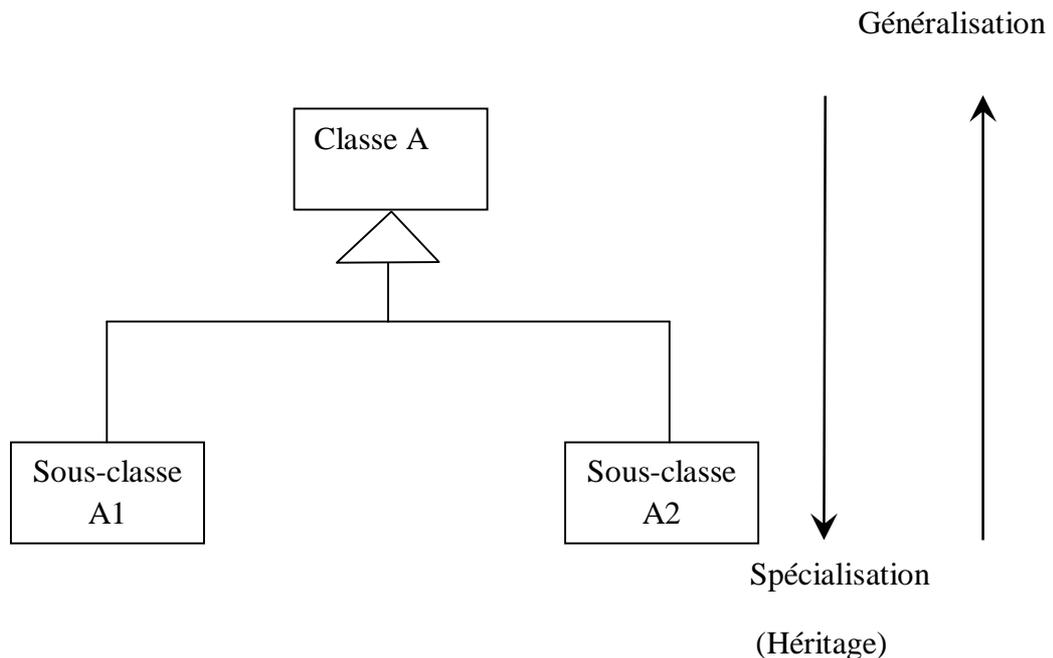
La généralisation est la relation entre une classe et deux autre classes ou plus partageant un sous ensemble commun d'attributs et/ou d'opérations.

La classe qui est affiliée s'appelle sur-classe, les classes affiliées s'appelle sous-classe.

L'opération qui consiste à créer une super-classe à partir de classes s'appelle la généralisation.

Inversement la spécialisation consiste à créer des sous-classes à partir d'une classe.

Formalisme :



a. L'héritage :

L'héritage permet à une sous classe de disposer des attributs et des opérations de la classe dont elle dépend. Un discriminateur peut être utilisé pour exploiter le critère de spécialisations entre une classe et ses sous classes. Le discriminant est simplement indiqué sur le schéma, puisque les valeurs prises par se discriminant correspondent à chaque sous classes.

b. Classe abstraite :

Une classe abstraite est une classe qui n'a pas d'instances directes mais dont les classes descendantes ont des instances. Dans une relation d'héritage, la super classe est par définition une classe abstraite.

c. L'héritage avec recouvrement :

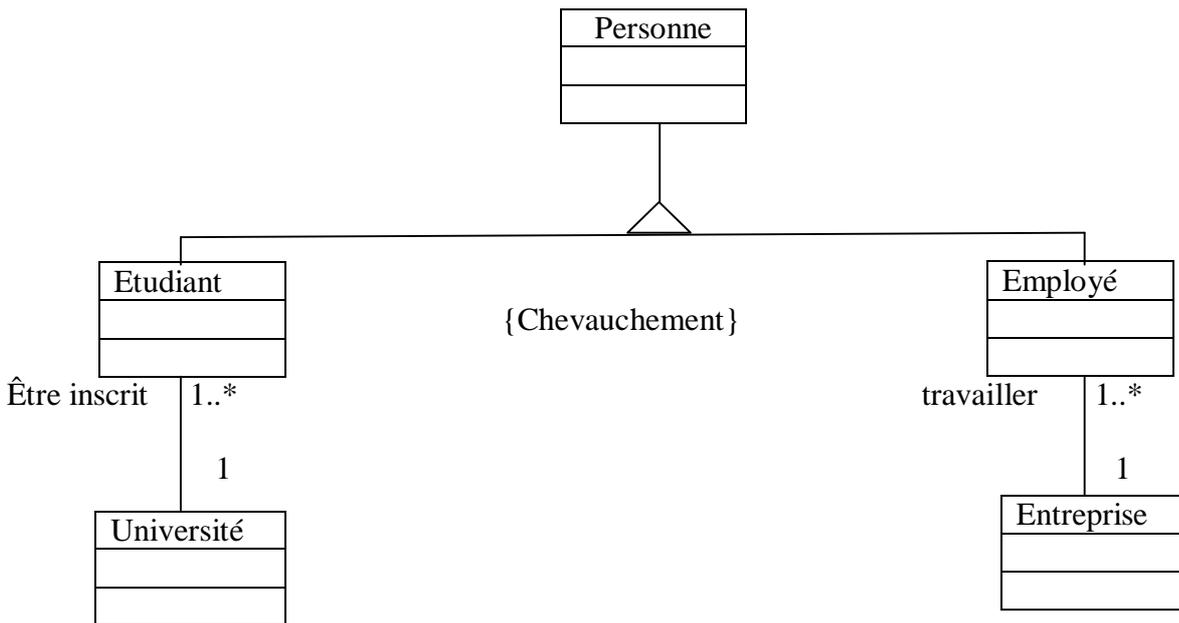
Par défaut, les sous classes ont des instances disjointes les unes par rapport aux autres. Dans certain cas, il existe un recouvrement d'instances entre les sous classes. D'une manière générale, quatre situation peuvent se rencontrer et se représentent sous forme de contraintes :

- {chevauchement} : deux sous classes peuvent avoir, parmi leur instances, des instances identiques ;
- {disjoint} : les instances d'une sous classe ne peuvent être incluses dans une autre sous classe de la même classe ;

- {complète} : la généralisation ne peut pas être étendue ;
- {incomplète} : la généralisation peut être étendue.

Dans certains cas, il est possible de ne pas citer toutes les sous classes mais d'indiquer seulement des points de suspension (...).

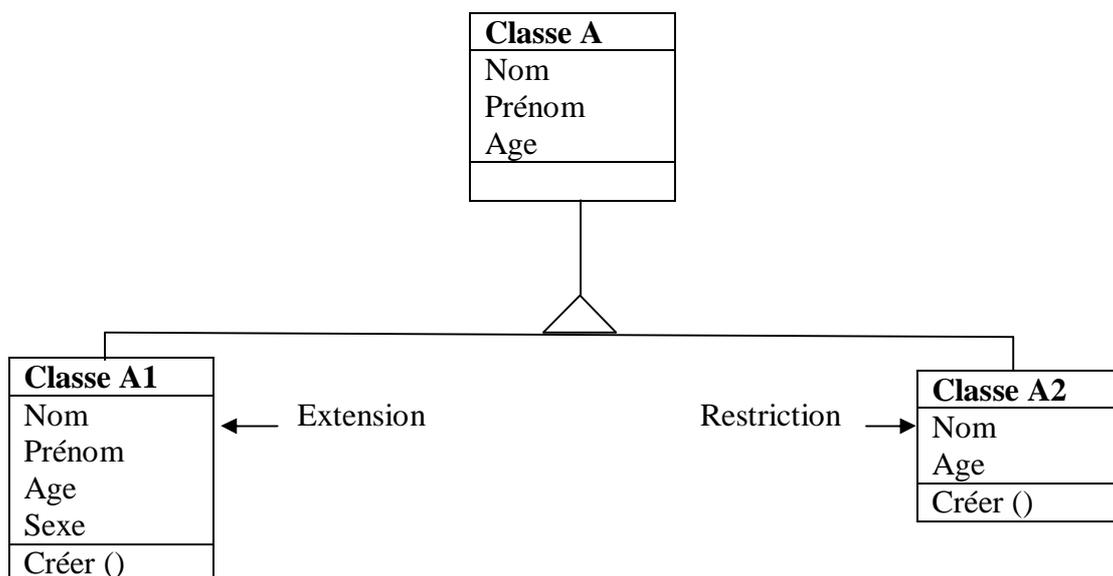
Formalisme :



d. Extension et restriction de classe :

L'ajout d'une propriété dans une sous classe correspond à une extension de classe. Le masquage d'une propriété dans une sous classe correspond à une restriction de classe.

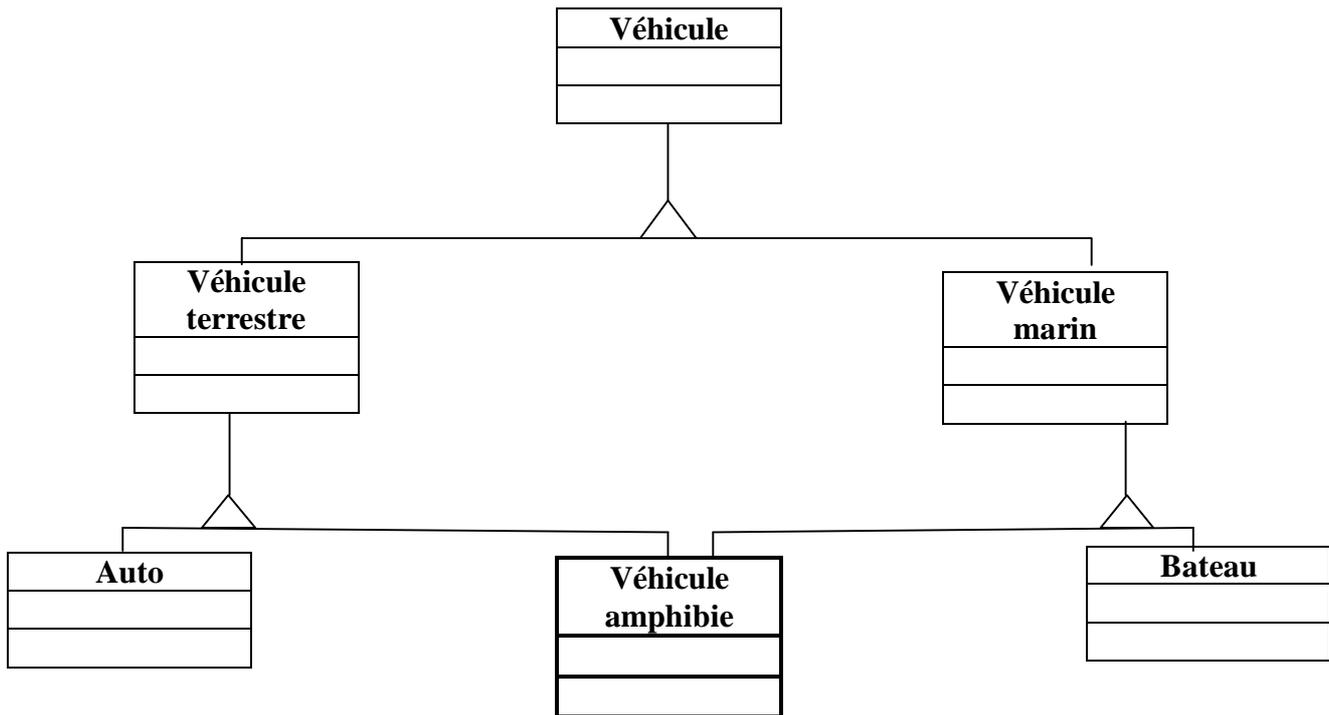
Formalisme :



e. L'héritage multiple :

Dans certains cas, il est nécessaire de faire hériter une même classe de deux classes « parentes » distinctes. Ce cas correspond à un héritage multiple.

Formalisme :



IV.1.6. Stéréotype de classe :

UML propose un certain nombre de stéréotype qui permettent de qualifier les profils d'utilisation.

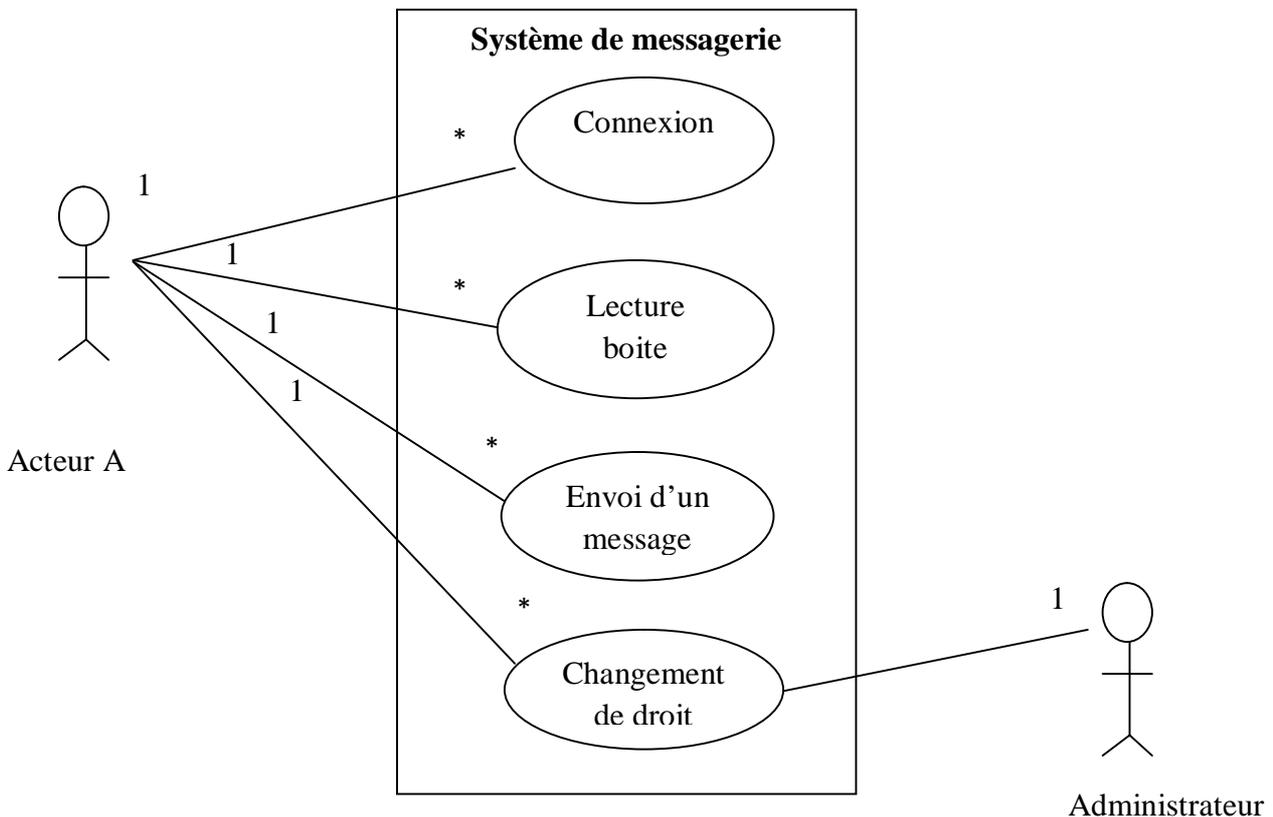
Parmi ces stéréotypes, nous présentons ci-après quatre d'entre eux :

- « Classe d'implémentation » : ce stéréotype est utilisé pour décrire des classes de niveau physique.
- « Type » : ce stéréotype permet de spécifier des opérations applicables à un domaine d'objets.
- « Utilitaire » : ce stéréotype qualifie toutes les fonctions utilitaires de base utilisées par les objets.
- « MétaClasse » : ce stéréotype permet de regrouper des classes dans une famille de classe.

IV.2. Diagramme des cas d'utilisation :

Tout système peut être décrit par un certain nombre de cas d'utilisation correspondant aux besoins exprimés par l'ensemble des utilisateurs. A chaque utilisateur, vu comme acteur, correspondra un certain nombre de cas d'utilisation du système.

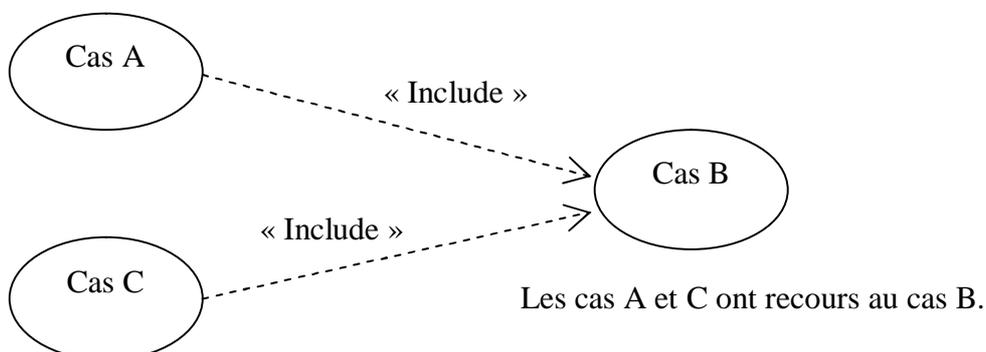
L'ensemble de ces cas d'utilisation se représente sous forme d'un diagramme.

Exemple d'un diagramme de cas d'utilisation :**IV.2.1. Relation entre cas d'utilisation :**

Afin d'optimiser la formalisation des besoins en ayant recours notamment à la réutilisation de cas d'utilisation, trois relations peuvent être décrites entre cas d'utilisation : une relation d'inclusion (« include »), une relation d'extension (« extend ») et une relation de généralisation.

a. Relation d'inclusion :

Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.

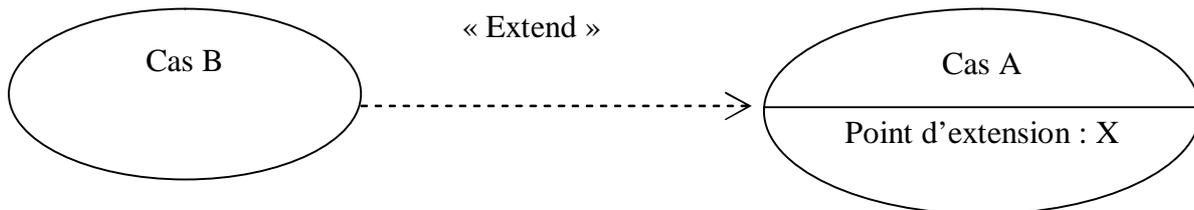
Formalisme :

b. Relation d'extension :

Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B. deux caractéristiques sont à noter :

- Le caractère optionnel de l'extension dans le déroulement du cas d'utilisation standard ;
- La mention explicite du point d'extension dans le cas d'utilisation standard.

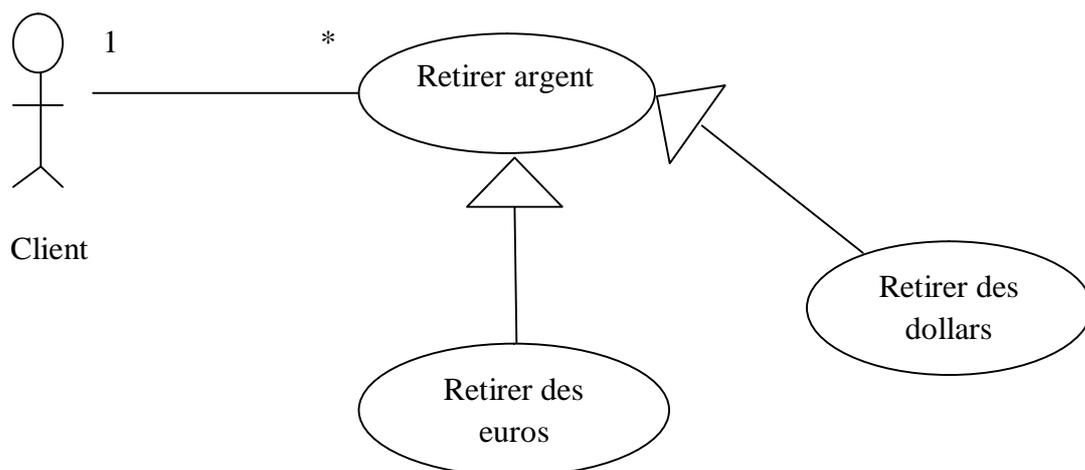
Formalisme :



c. Relation de généralisation :

Une relation de généralisation de cas d'utilisation peut être défini conformément au principe de la spécialisation-généralisation déjà présentée pour les classes.

Formalisme : exemple d'une relation de généralisation de cas d'utilisation :



IV.2.2. Description textuelle d'un cas d'utilisation :

A chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs.

La description textuelle d'un cas d'utilisation est articulée en six points :

- Objectif : Décrire succinctement le contexte et les résultats attendus du cas d'utilisation.
- Acteurs concernés : Le ou les acteurs concernés par le cas doivent être identifiés en précisant globalement leur rôle.
- Pré condition : Si certaines conditions particulières sont requises avant l'exécution du cas, ils sont à expliquer à ce niveau.
- Post condition : Par symétrie, si certaines conditions particulières doivent être réunies après l'exécution du cas, elles sont à exprimer à ce niveau.
- Scénario nominal : Il s'agit là du scénario principale qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaiter.
- Scénario alternatif : Les autres scénarios secondaires ou correspondants à la résolution d'anomalies, sont à décrire à ce niveau. Le lien avec le scénario principal se fait à l'aide d'une numérotation hiérarchisée (1.1a, 1.1b...) rappelant le numéro de l'action concernée.

IV.3. Diagramme d'activité :

Le diagramme d'activité présente un certain nombre de points communs avec le diagramme d'état-transition puisqu'il concerne le comportement interne des opérations ou des cas d'utilisation.

Les concepts communs ou très proches entre le diagramme d'activité et le diagramme d'état-transition sont :

- Transition,
- ● Nœud initial (état initial),
- ⊙ Nœud final (état final),
- ⊗ Nœud de fin flot (état de sortie),
- ◇ Nœud de décision (choix).

IV.3.1. Concepts du diagramme d'activité :

Les concepts spécifiques au diagramme d'activité sont :

- Nœud de bifurcation,
- Nœud de jonction,
- Nœud de fusion,
- Pin d'entrée et de sortie,
- Flot d'objet,
- Partition.

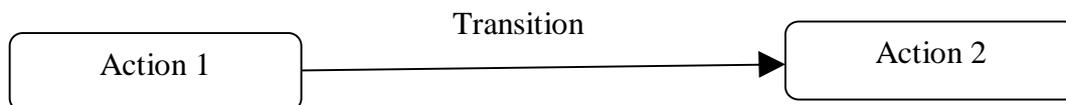
a. Action :

Une action correspond à un traitement qui modifie l'état du système. Cette action peut être appréhendée soit à un niveau élémentaire proche d'une instruction en termes de programmation soit à un niveau plus globale correspondant à une ou plusieurs opérations.

Formalisme :**b. Transition et flot de contrôle :**

Dès qu'une action est achevée, une transition automatique est déclenchée vers l'action suivante. Il n'y a donc pas d'événement associé à la transition.

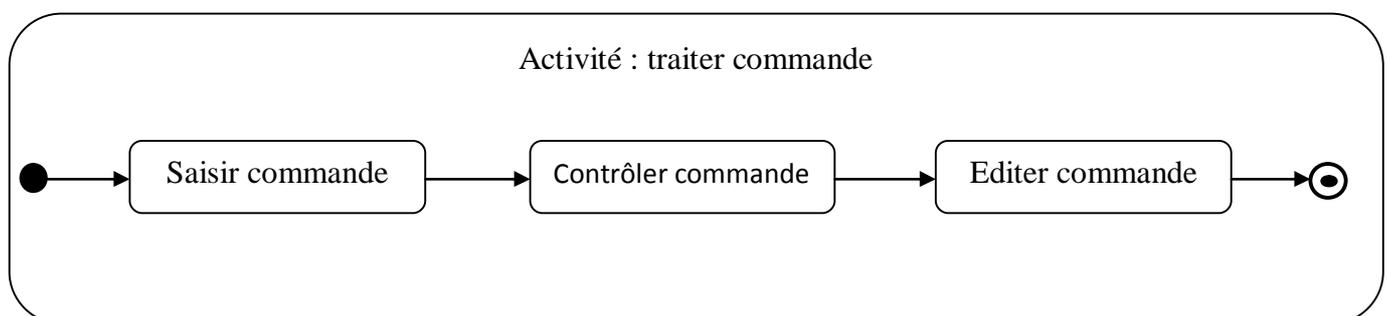
L'enchaînement des actions constitue le flot de contrôle.

Formalisme :**c. Activité :**

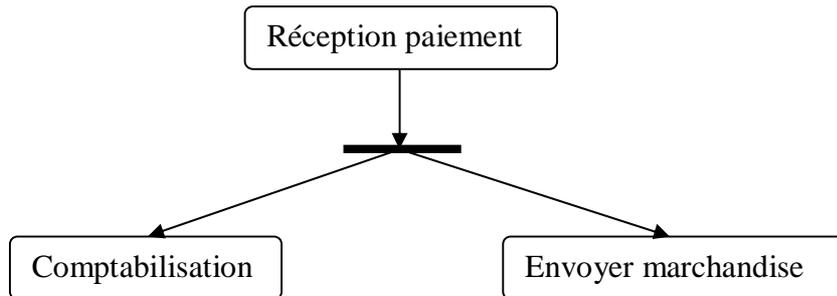
Une activité représente le comportement d'une partie du système en termes d'actions et de transitions. Une activité est composée de trois types de nœuds :

- Nœud d'exécution (action, transition),
- Nœud de contrôle (nœud initial, nœud final, flux de sortie, nœud de bifurcation, nœud de jonction, nœud de fusion-test, nœud de test-décision, pin d'entrée et sortie),
- Nœud d'objet.

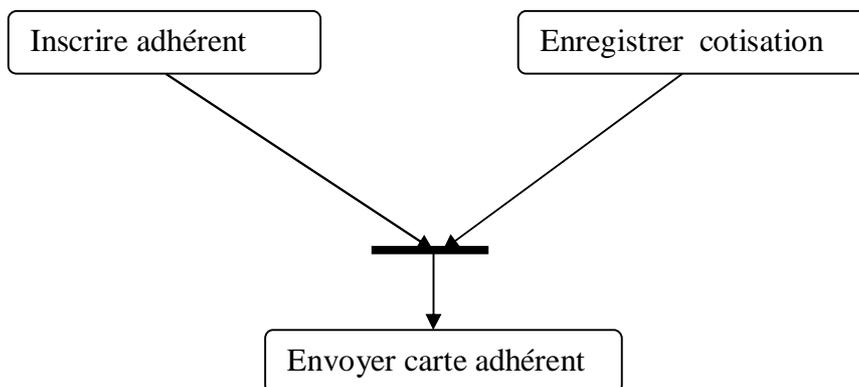
Une activité peut recevoir les paramètres en entrée et en produire en sortie.

Formalisme :**d. Nœud de bifurcation (fourche) :**

Un nœud de bifurcation permet à partir d'un flot unique entrant de créer plusieurs flots concurrents en sortie de la barre de synchronisation.

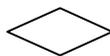
Formalisme :**e. Nœud de jonction (synchronisation) :**

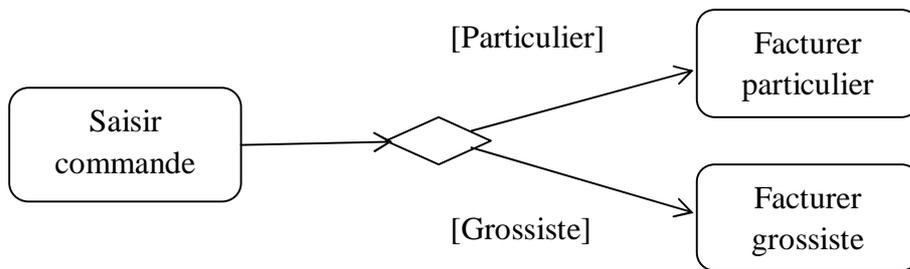
Un nœud de jonction permet, à partir de plusieurs flots concurrents en entrée de la synchronisation, de produire un flot unique sortant. Le nœud de jonction est le symétrique du nœud de bifurcation.

Formalisme :**f. Nœud de test-décision :**

Un de test-décision permet de faire un choix entre plusieurs flots sortant en fonction des conditions de garde de chaque flot. Un nœud de test décision n'a qu'un seul flot en entrée. On peut aussi utiliser seulement deux flots de sortie : le premier correspondant à la condition vérifiée et l'autre traitant le cas sinon.

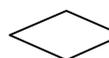
Formalisme :

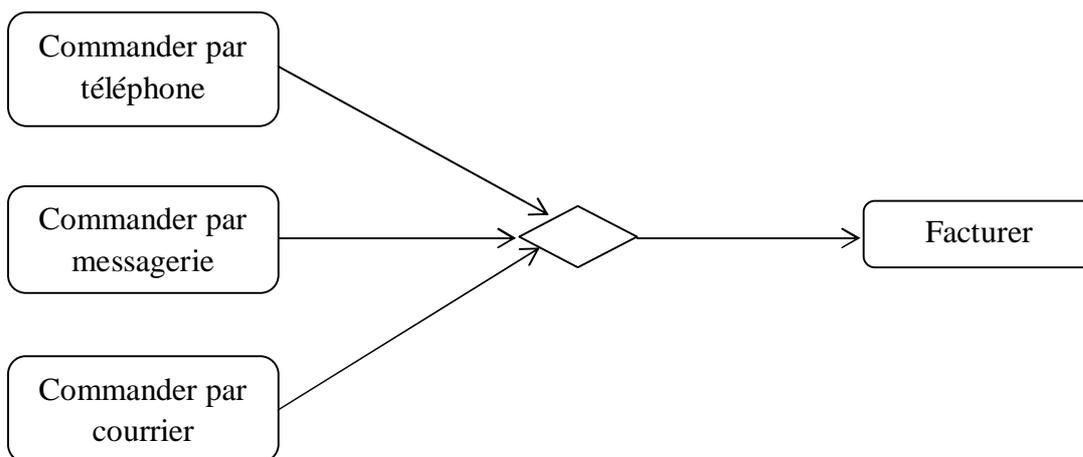
 Symbole du nœud de test-décision

**g. Nœud fusion-test :**

Un nœud de fusion-test permet d'avoir plusieurs flots entrant possibles et un flot sortant. Le flot sortant est donc exécuter dès qu'un dès flot entrants est activé.

Formalisme :

 Nœud de fusion-test

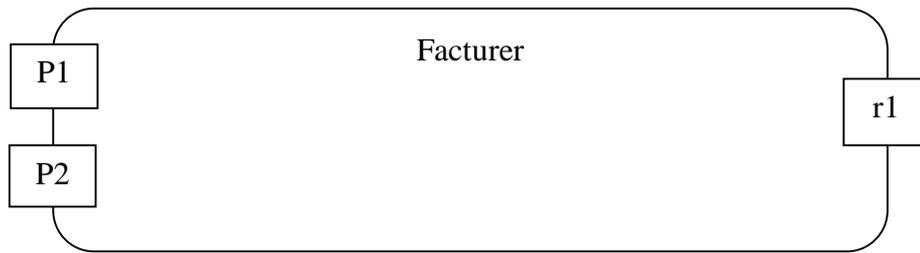
**h. Pin d'entrée et de sortie :**

Un pin d'entrée ou de sortie représente un paramètre que l'on peut spécifier en entrée ou en sortie d'une action. Un nom de donnée et un type de donnée peuvent être associé au pin . un paramètre peut être de type objet.

Formalisme :

Chaque paramètre se représente dans un petit rectangle. Le nom du paramètre et son type sont aussi à indiquer.

 Pin d'entrée ou de sortie.



P1 : entier.

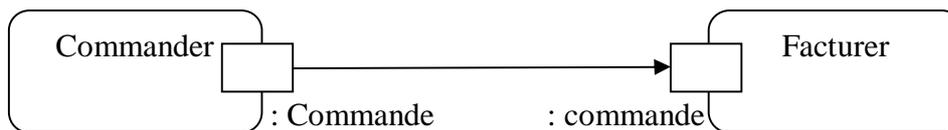
P2 : texte.

r1 : réel.

i. Flot de données et nœud d'objet :

Un nœud d'objet permet de représenter le flot de données véhiculé entre les actions. Les objets peuvent se représenter de deux manières différentes : soit en utilisant le pin d'objet soit en représentant explicitement un objet.

Formalisme :



Partition :

UML permet aussi d'organiser la présentation du diagramme d'activité en couloir d'activité. Chaque couloir correspond à un domaine de responsabilité d'un certain nombre d'actions.

Les flots d'objets sont aussi représentés dans le diagramme. L'ordre relatif des couloir de responsabilité n'est pas significatif.

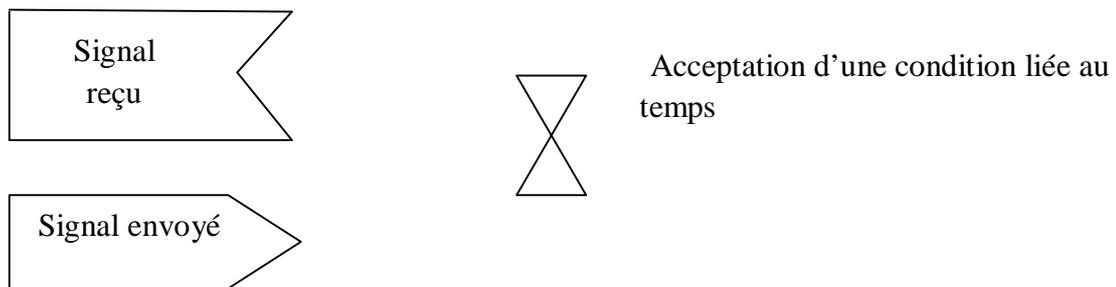
IV.3.2. Représentation d'action de communication :

Dans un diagramme d'activité, comme dans un diagramme de temps, des interactions de communication liées à certain types d'événements peuvent se représenter.

Les types d'événements concernés sont :

- Signal,
- Ecoulement du temps.

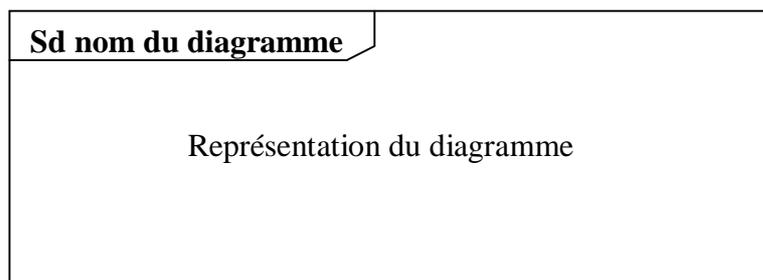
Formalisme :



IV.4. Diagramme de séquence :

L'objectif du diagramme de séquence est de représenter les interactions entre objet en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

Un diagramme de séquence se représente globalement comme suit :



IV.4.1. Ligne de vie :

Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite ou explicite à l'aide d'un message de retour.

IV.4.2. Opérations particulières :

a. Création et destruction d'objet :

Si un objet est créé par une opération, celui-ci n'apparaît qu'au moment où il est créé. Si l'objet est détruit par une opération, la destruction, se représente par « X ».

b. Contraintes temporelles :

D'abord des contraintes de chronologie entre les messages peuvent être spécifiées. De plus lorsque l'émission d'un message requière une certaine durée, il se représente sous forme d'un trait oblique.

IV.4.3. Fragment d'interaction :

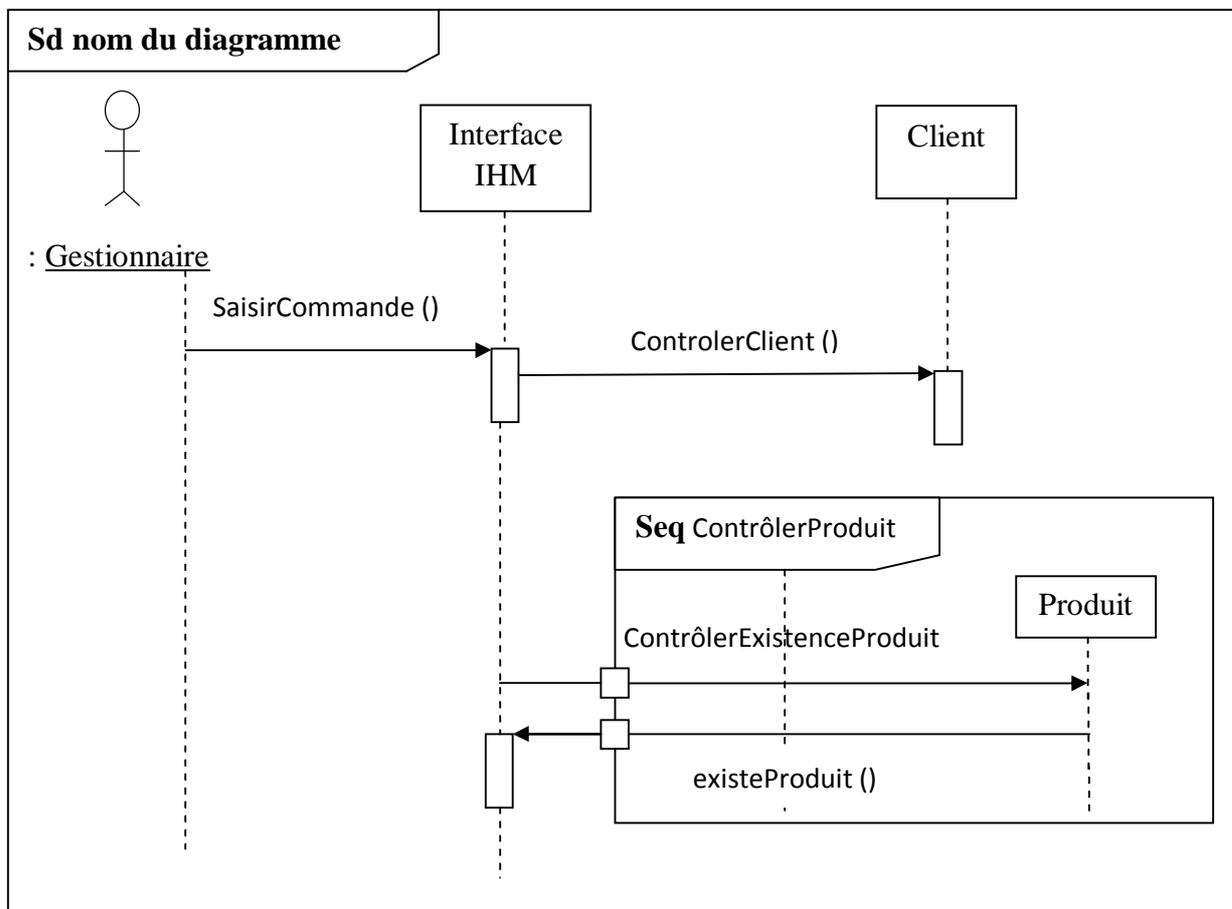
Types de fragments d'interaction :

Dans un diagramme de séquence, il est possible de distinguer des sous-ensembles d'interactions qui constituent des fragments.

Un fragment d'interaction se représente globalement dans un rectangle avec indication dans le coin à gauche du nom du fragment.

Un port d'entrée ou un port de sortie peuvent être indiqués pour connaître la manière dont ce fragment peut être relié au reste du diagramme. Dans le cas où aucun port n'est indiqué c'est l'ensemble du fragment qui est appelé pour exécution.

Formalisme :



Un fragment d'interaction dit combiné correspond à un ensemble d'interaction auquel on applique un opérateur. Un fragment combiné se représente globalement comme un diagramme de séquence avec indication dans le coin à gauche du nom de l'opérateur.

IV.4.4. Description des opérateurs :

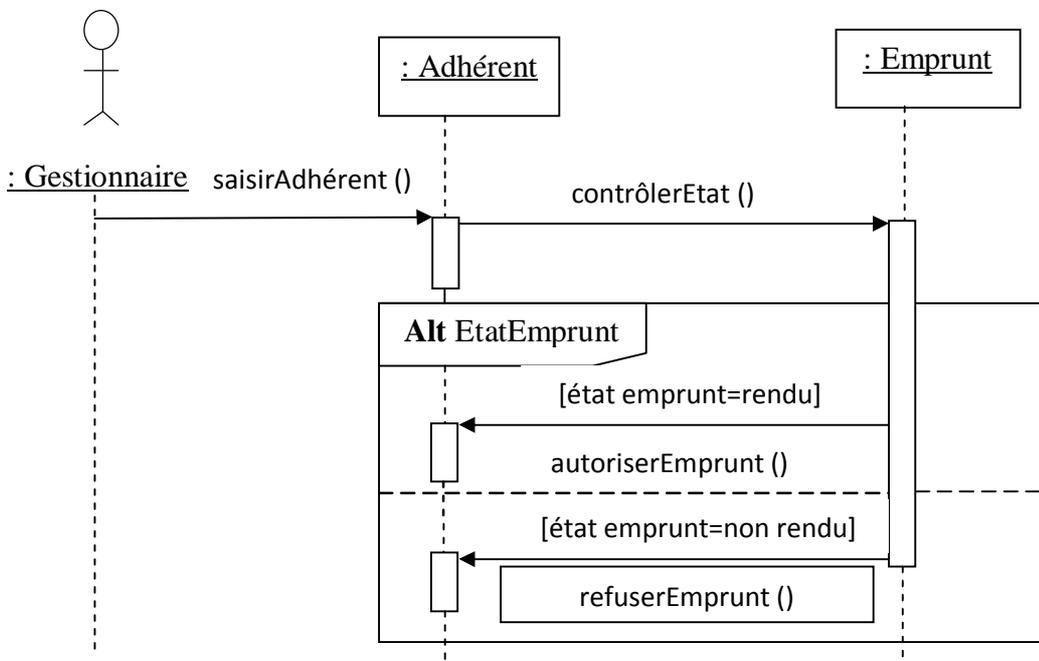
Treize opérateurs ont été définis dans UML :

a. Opérateur alt :

L'opérateur **alt** correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser des clauses de types sinon.

Formalisme :

L'opérateur **alt** se représente dans un fragment possédant aux moins deux parties séparées par des pointillés. L'exemple suivant montre cela :

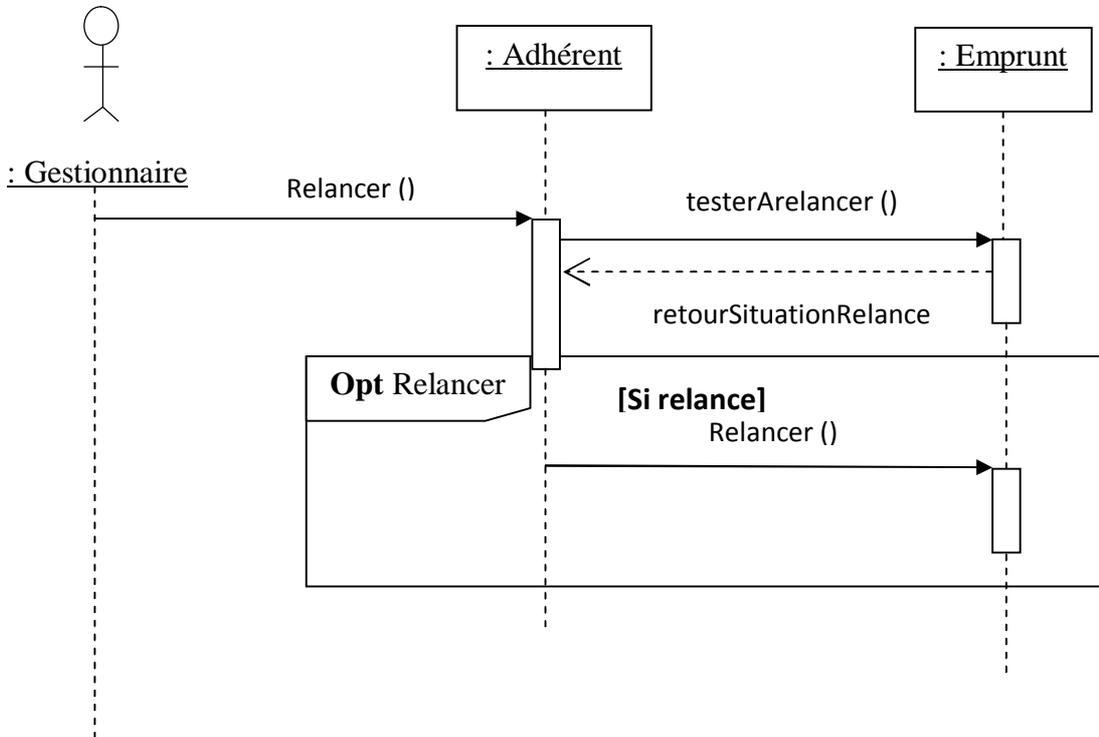


b. Opérateur opt :

L'opérateur **opt** (optional) correspond à une instruction de test sans alternative (sinon).

Formalisme :

L'opérateur **opt** se représente dans un fragment possédant une seule partie.



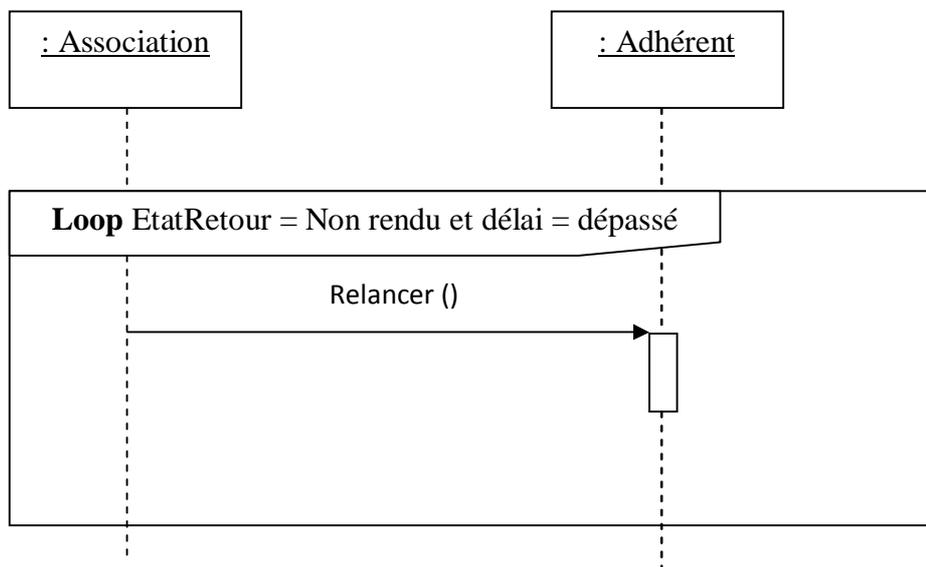
c. Opérateur loop :

L'opérateur **loop** correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite.

Il est possible aussi d'utiliser une condition portant sur un nombre minimum et maximum d'exécution de la boucle en écrivant : loop min, max. dans ce cas, la boucle s'exécutera au minimum min fois et au maximum max fois. Il est aussi possible de combiner l'option min/max avec la condition associée à la boucle.

Formalisme :

L'opérateur **loop** se représente dans un fragment possédant une seule partie et englobant toutes les interactions faisant partie de la boucle.



d. Opérateur par :

L'opérateur **par** (parallel) permet de représenter deux séries d'interactions qui se déroulent en parallèle.

Formalisme :

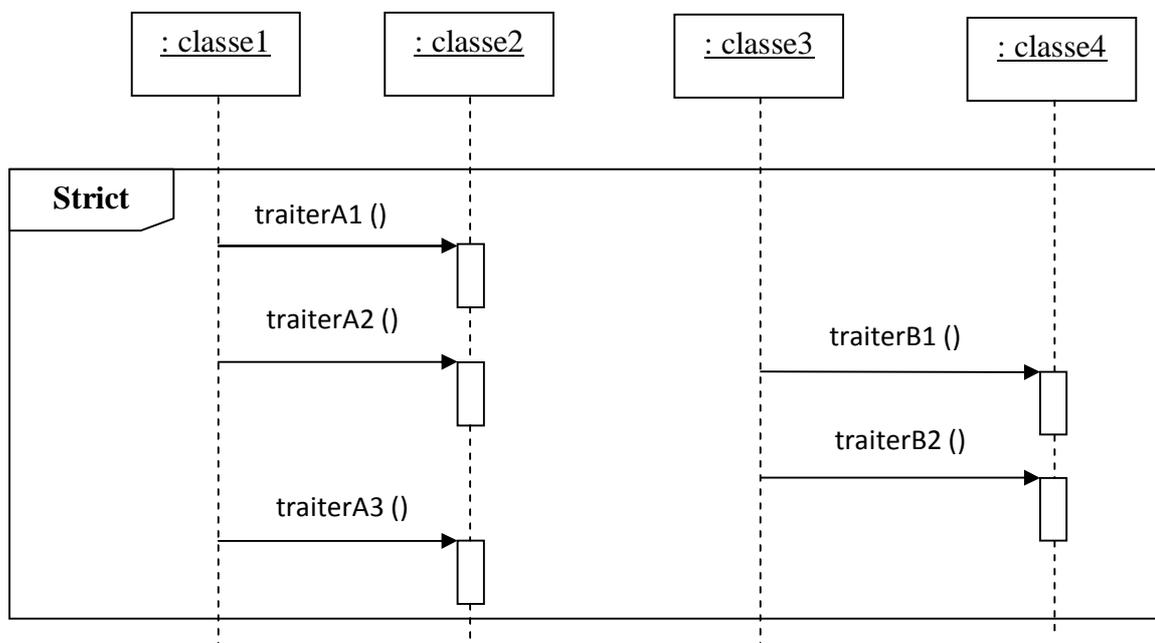
L'opérateur se représente dans un fragment possédant deux parties séparées par une ligne en pointillé. C'est un opérateur qui est plutôt utilisé dans l'informatique temps réel.

e. Opérateurs strict et weak sequencing :

Les opérateurs strict et weak permettent de représenter une série d'interactions dont certaines s'opèrent sur des objets indépendants :

- L'opérateur strict est utilisé quand l'ordre d'exécution des opérations doit être strictement respecté.
- L'opérateur weak est utilisé quand l'ordre d'exécution des opérateurs n'a pas d'importance.

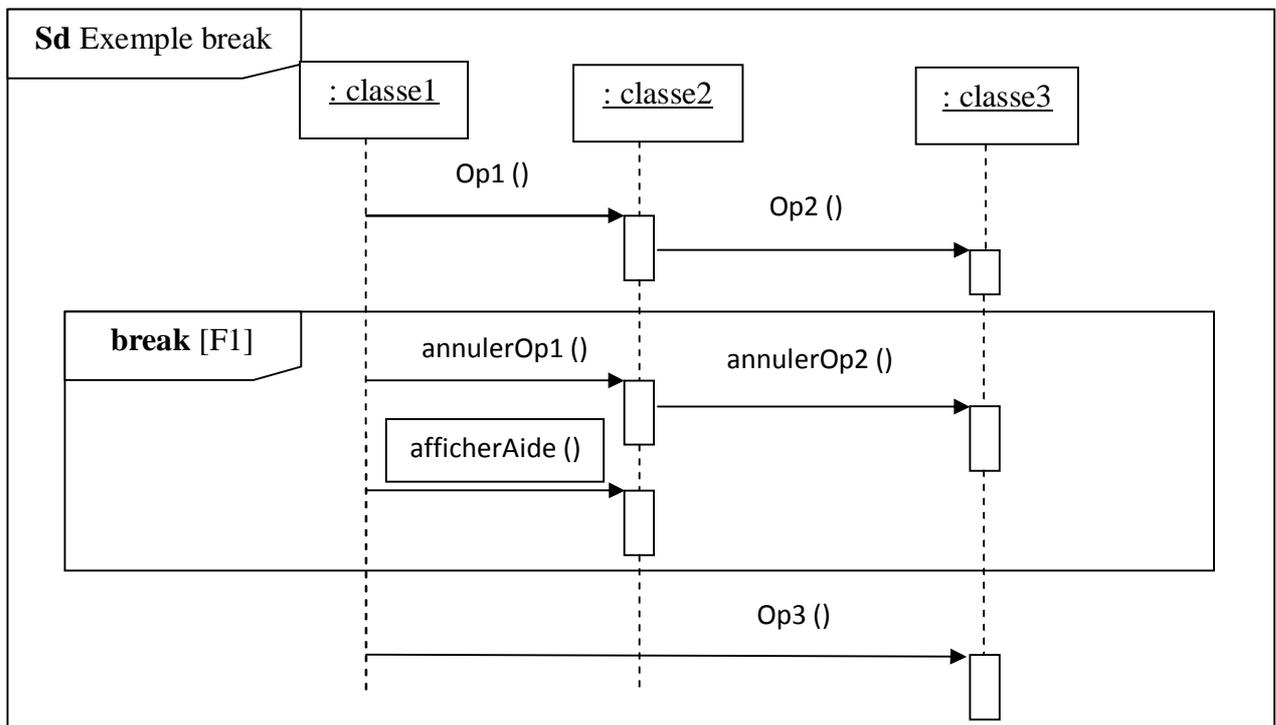
Formalisme :



Cet exemple montre que les opérations A1,A2,B1,B2 et A3 doivent être exécutées dans cet ordre.

f. Opérateur break :

L'opérateur break permet de représenter une situation exceptionnelle correspondant à un scénario de rupture par rapport au scénario général. Le scénario de rupture s'exécute si la condition de garde est satisfaite.

Formalisme :

Cet exemple montre que les opérations annulerOp1 (), annulerOp2 () et afficheAide () ne seront exécutées que si la touche F1 est activée sinon le fragment est ignoré et la séquence de traitement passe directement de l'opération Op2 () à l'opération Op3 ().

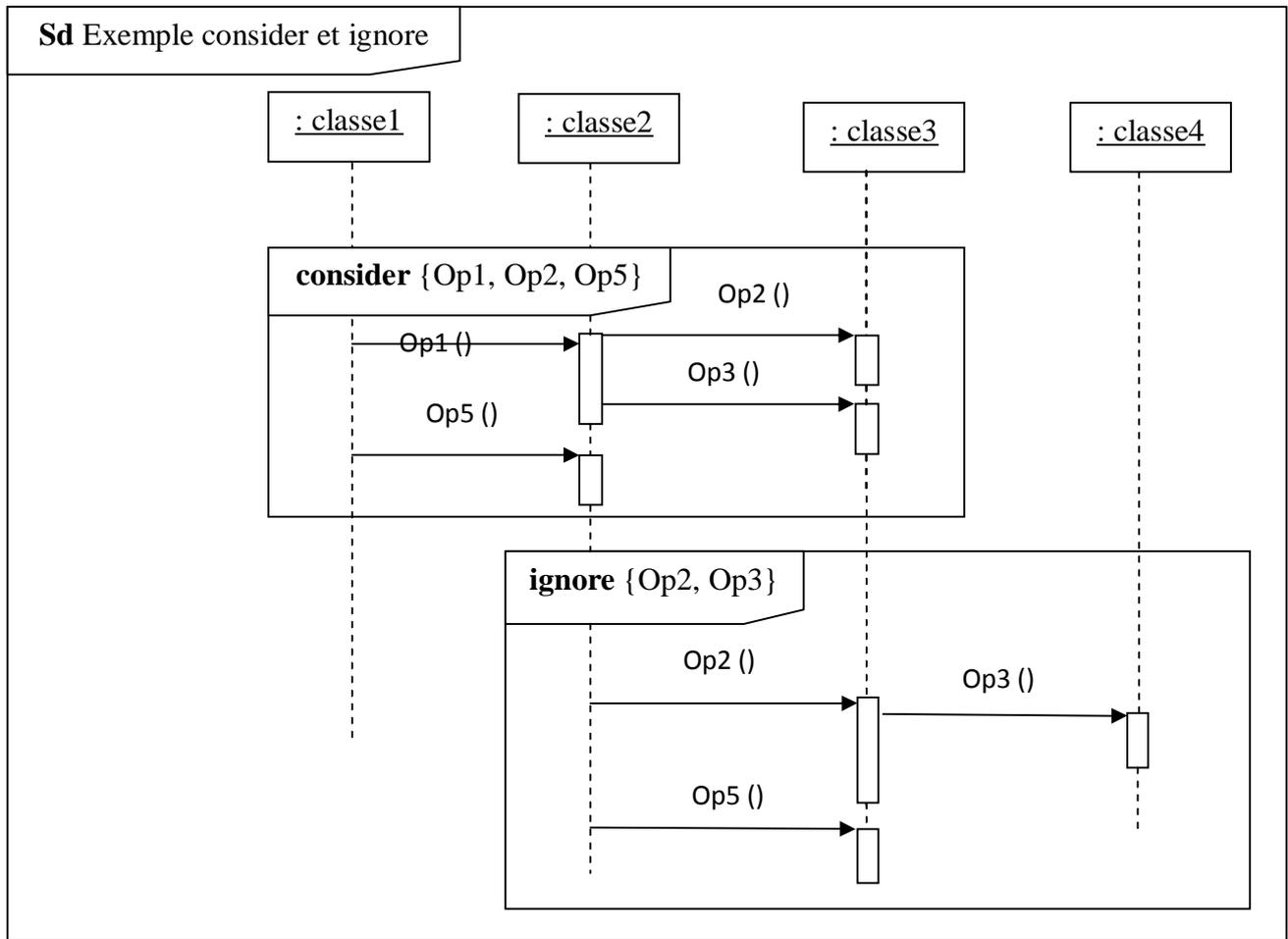
g. Opérateurs ignore et consider :

Les opérateurs ignore et consider sont utilisés pour des fragments d'interactions dans lesquels on veut montrer que certains messages peuvent être soit absents sans avoir d'incidence sur le déroulement des interactions (ignore), soit obligatoirement présents (consider).

Formalisme :

Dans l'exemple suivant on montre :

- Dans le fragment consider, les messages Op1, Op2 et Op5 doivent être obligatoirement présent lors de l'exécution du fragment sinon le fragment n'est pas exécuté,
- Dans le fragment ignore, les messages Op2 et Op3 peuvent être absents lors de l'exécution du fragment.

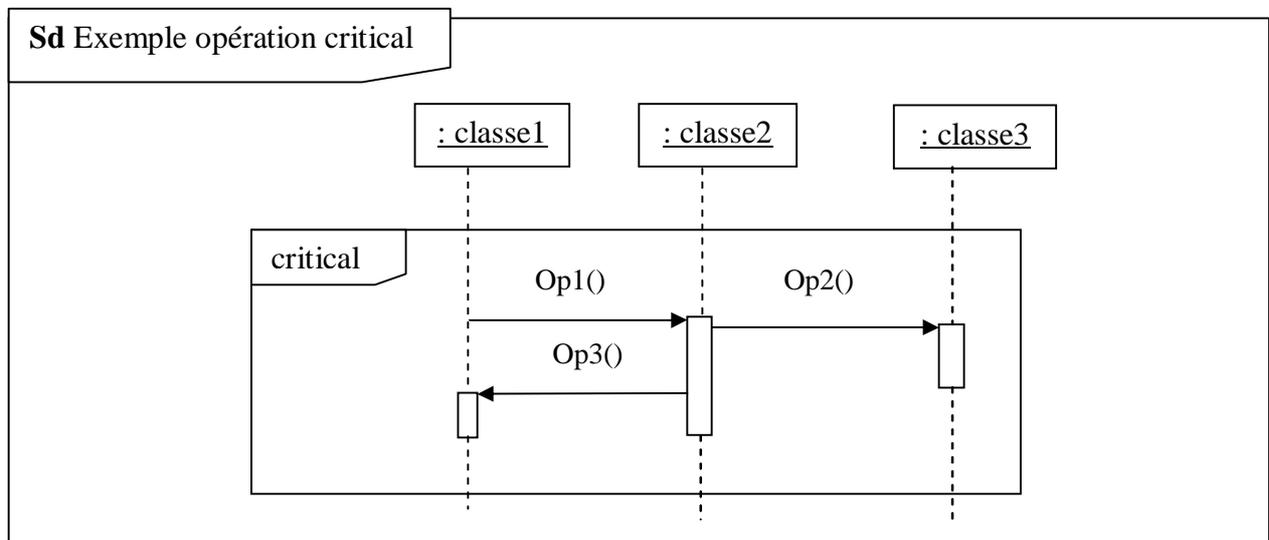


h. Opérateur critical :

L'opérateur **critical** permet d'indiquer qu'une séquence d'interactions ne peut être interrompue compte tenu du caractère critique des opérations traitées. On considère que le traitement des interactions comprises dans la séquence critique est atomique.

Formalisme :

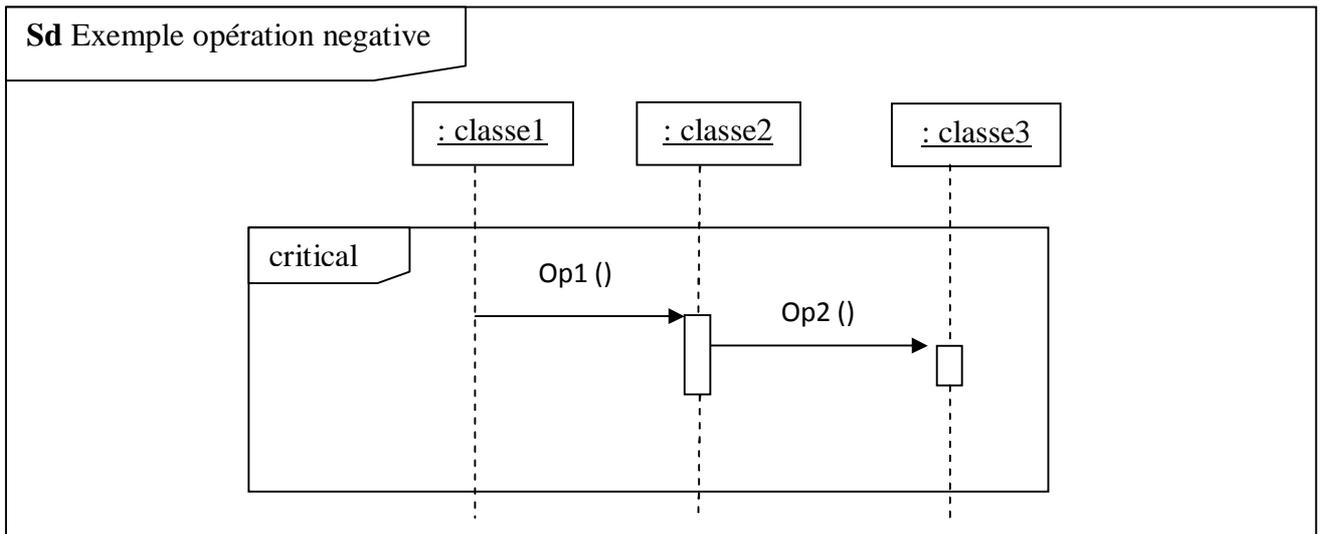
L'exemple montre que les opérations Op1 (), Op2 () et Op3 () du fragment critical doivent s'exécuter sans interruption.



i. Opérateur négative :

L'opérateur **neg** (negative) permet d'indiquer qu'une séquence d'interactions est invalide.

Formalisme :

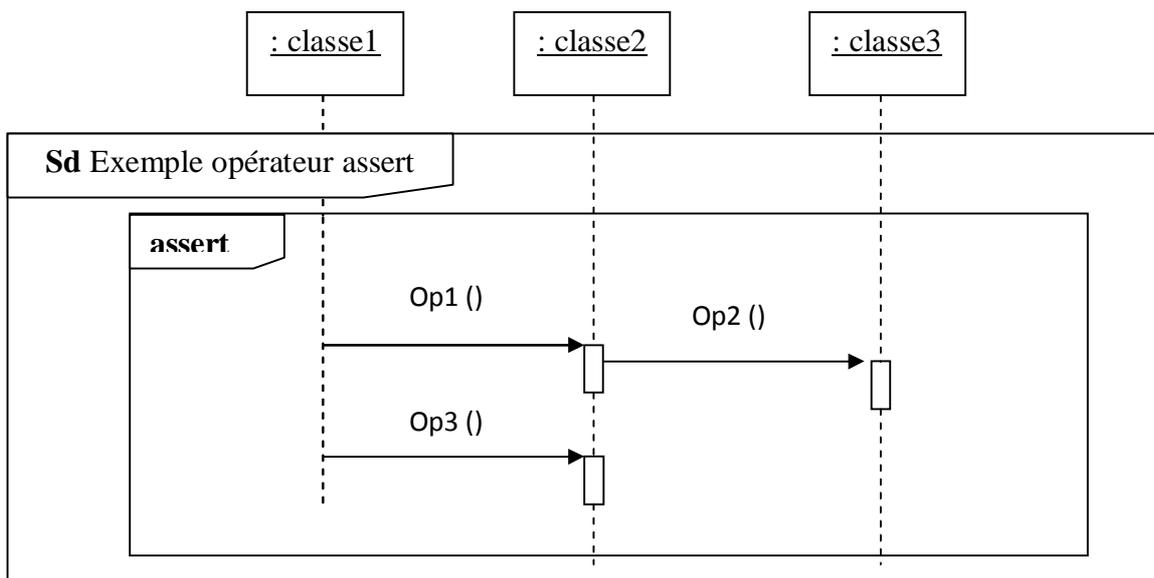


Les opérations Op1 () et Op2 () du fragment neg sont invalides. Une erreur sera déclenchée dans ce cas à l'exécution du fragment.

j. Opérateur assertion :

L'opérateur **assert** (assertion) permet d'indiquer qu'une séquence d'interactions est l'unique séquence possible en considérant les messages échangés dans le fragment. Toute autre configuration de message est invalide.

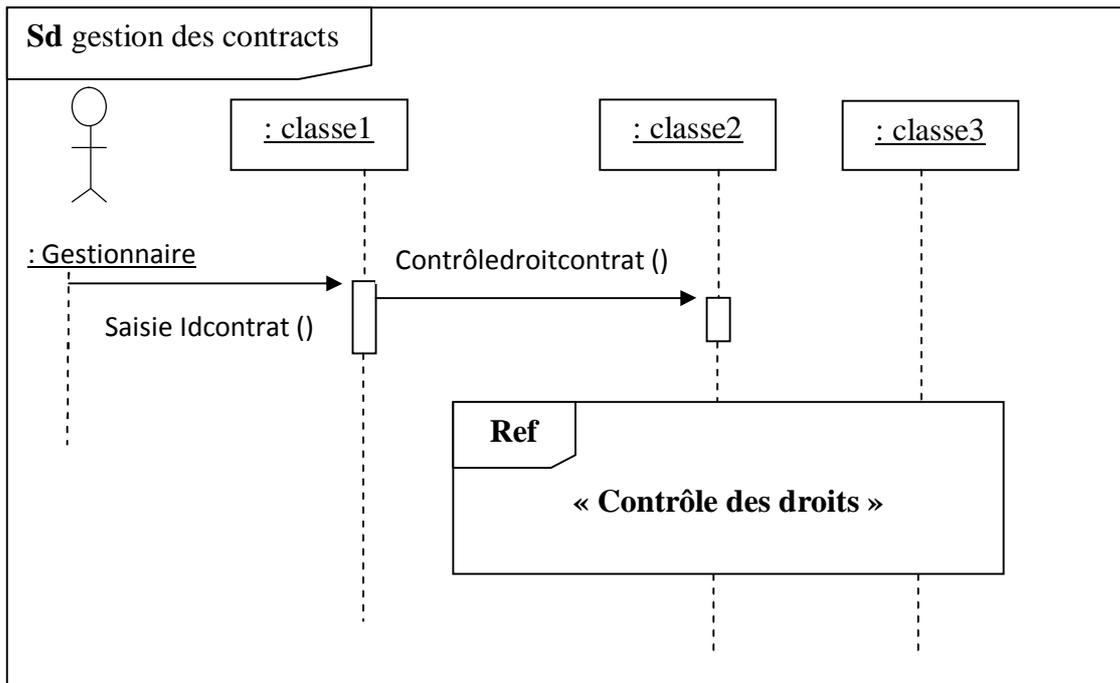
Formalisme :



Le fragment assert ne s'exécutera que si l'unique séquence de traitement Op1 (), Op2 () et Op3 () se réalise

k. Opérateur ref :

L'opérateur ref permet d'appeler une séquence d'interactions décrite pzf ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

Formalisme :

Introduction :

L'informatique d'entreprise, autrefois dominée par les systèmes centraux (mini-ordinateur et grand système,) a été bouleversée par le développement de la micro informatique. Celle-ci a apporté à ses utilisateurs et aux informaticiens des outils efficaces, plus souples, plus confortables et surtout moins coûteux.

Après une phase d'utilisation « individualiste » de la micro informatique, l'adoption généralisée des réseaux locaux a permis de mieux organiser les informations dispersées, d'améliorer la communication entre les utilisateurs et de réduire les dépenses par le partage des périphériques et des applications.

I. Réseaux informatiques :

I.1. Définition : Ensemble d'ordinateurs reliés entre eux grâce à des lignes physiques et échangeant des informations sous forme de données numériques.

Un réseau informatique peut servir plusieurs buts distincts :

- Le partage de ressources (fichiers, applications ou matériels, connexion à internet, etc.) ;
- La communication entre personnes (courrier électronique, etc.) ;
- La communication entre processus (entre des ordinateurs industriels par exemple) ;
- La garantie de l'unicité et de l'universalité de l'accès à l'information (bases de données en réseau) ;

I.2. Type de réseaux : [W08]

Les réseaux sont généralement classifiés suivant le type de liaisons utilisées (la topologie) et la surface géographique qu'ils couvrent.

I.3. Architecture des réseaux :

Les réseaux sont structurés du point de vue fonctionnel en deux catégories :

- Les réseaux Poste à Poste (Peer to Peer).
- Les réseaux organisés au tour de serveur (client / serveur).

I.3.1. Architecture d'égal à égal (peer to peer): [B09]

Contrairement à une architecture de réseau de type client/serveur, dans une architecture d'égal à égal (peer to peer), il n'y a pas de serveur dédié. Chaque ordinateur est à la fois client et serveur. Le "égal à égal" ou encore poste à poste représente une alternative plus simple et moins chère au réseau classique, de type Novell ou Windows NT.

Il est donc libre de partager ses ressources (disques, connexion Internet...)

✓ Avantage :

- Un coût réduit : Les coûts engendrés par un tel réseau sont d'ordre matériel (postes de travail + cartes réseau + câbles) ;
- Faible maintenance ;
- Une simplicité à toute épreuve ;

✓ Inconvénients :

- Ce système n'est pas du tout centralisé, ce qui le rend difficile à administrer ;
- La sécurité est très peu présente ;
- les réseaux d'égal à égal ne sont valables que pour un petit nombre d'ordinateurs (généralement une dizaine), et pour des applications ne nécessitant pas une grande sécurité ;

I.3.2. L'architecture client/serveur : [B10]**Définition :**

Le client serveur est avant tout un mode de dialogue entre deux processus. Le premier appelé client demande l'exécution de services au second appelé serveur. Le serveur accomplit les services et envoie en retour des réponses. En général, un serveur est capable de traiter les requêtes de plusieurs clients. Un serveur permet donc de partager des ressources entre plusieurs clients qui s'adressent à lui par des requêtes envoyées sous forme de messages.

Ø Caractéristiques d'un serveur :

Le serveur est considéré comme le centre d'un réseau. C'est le cerveau du réseau.

- § Il est initialement passif (ou esclave, en attente d'une requête) ;
- § Il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients ;
- § Dès qu'une requête lui parvient, il la traite et envoie une réponse ;
- § Il est composé des mêmes sous-ensembles qu'un ordinateur standard, mais ces sous-ensembles sont beaucoup mieux optimisés ;
- § Il contient plus de mémoire vive ;
- § Son contrôleur de disques est de très bonne qualité (SCSI, voire Wide ou Ultra Wide SCSI) ;
- § Disques durs de très grande capacité ;
- § Microprocesseur(s) de dernière génération ;
- § Capacités de gestion de réseau.

Ø Caractéristiques d'un client :

Un client est capable de traiter des informations qu'il récupère auprès du serveur :

- § Il est actif en premier (ou maître).
- § Il envoie des requêtes au serveur.
- § Il attend et reçoit les réponses du serveur.

I.3.2.1. Fonctionnement d'un système client/serveur :

Un système client/serveur fonctionne selon le schéma suivant :

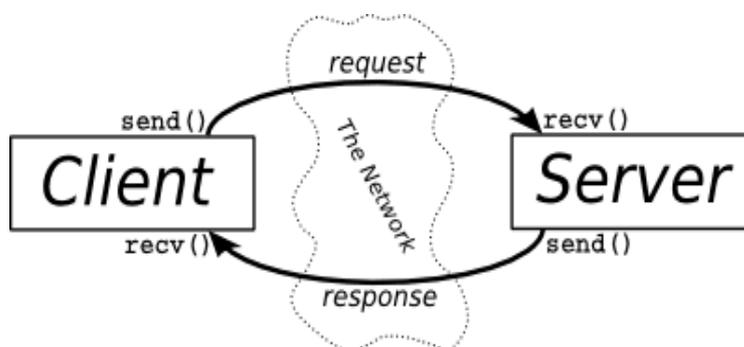


Figure I.1.15 : Système client/serveur

- Ü Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur ;
- Ü Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port ;

✓ Avantage :

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.
- *Une meilleure sécurité* : car le nombre de points d'entrée permettant l'accès aux données est moins important.
- **Une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

▼ Inconvénients :

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

- Un coût élevé dû à la technicité du serveur.
- Un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui. Le serveur a une grande tolérance aux pannes (notamment grâce au système RAID).

Remarque : RAID (Redundant Array of Independent/Inexpensive Disks, c'est-à-dire un groupe de disques redondants et indépendants /bon marché) désigne une architecture matérielle (parfois logiciel) permettant d'accélérer, de sécuriser et/ou de fiabiliser des accès aux données stockées sur les disques durs. Cette architecture est basée sur la multiplication des disques durs, par opposition à la méthode SLED (Single Large Expensive Disk) où toutes les données sont rassemblées sur un seul disque de prix élevé.

I.3.2.2. Les différentes architectures client/serveur : [B11]

a) L'architecture à 2 niveaux :

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

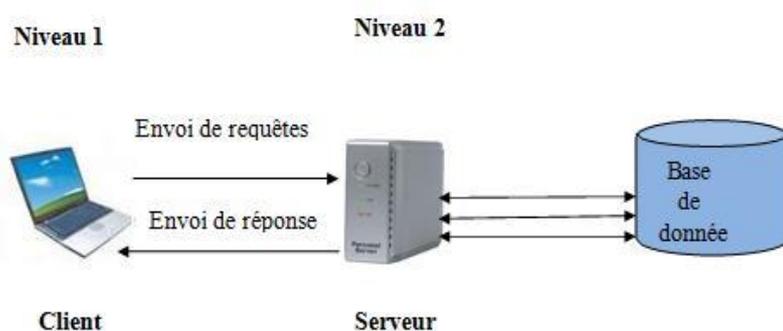


Figure I.1.16 : L'architecture à deux niveaux

b) L'architecture à 3 niveaux :

Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

1. Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
2. Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur.
3. Le serveur de données, fournissant au serveur d'application les données dont il a besoin.

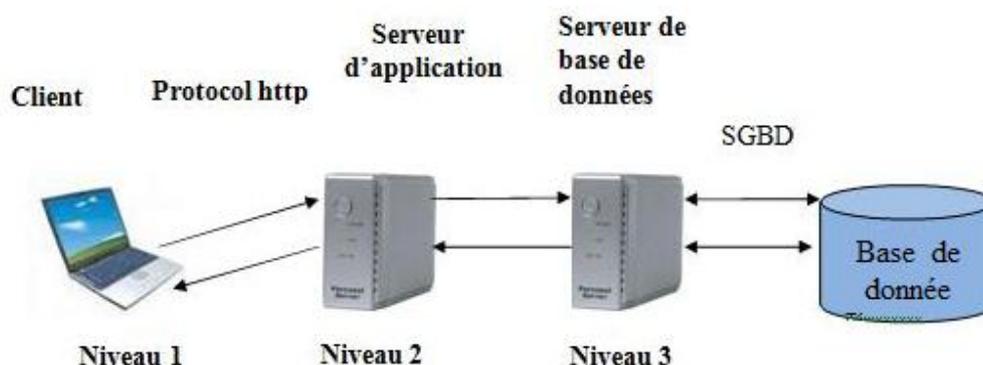


Figure I.1.17 : L'architecture à trois niveaux.

Le middleware : Est le complément de services du réseau permettant la réalisation du dialogue client/serveur, il prend en compte les requêtes de l'application cliente, les transmet de manière transparente à travers le réseau jusqu'au serveur et prend en compte les données résultants du Serveur vers l'application.

ü L'objectif essentiel du middleware est d'offrir aux applications une interface unifiée permettant l'accès à l'ensemble des services disponibles sur le réseau API.

Remarque :

Etant donné l'emploi massif du terme d'architecture à 3 niveaux, celui-ci peut parfois désigner aussi les architectures suivantes :

- Partage d'application entre client, serveur intermédiaire, et serveur d'entreprise ;
- Partage d'application entre client, serveur d'application, et serveur de base de données d'entreprise qui est l'architecture utilisée dans notre application (serveur d'application « Apache », serveur de base de données « MySQL » et le client « Navigateur »).

c) L'architecture multi niveaux :

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux...

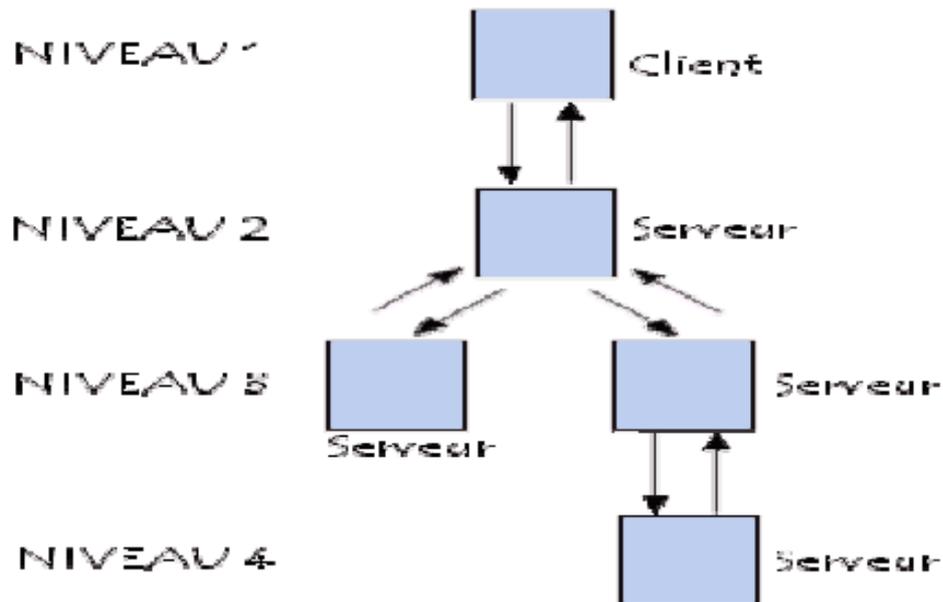


Figure I.1.18 : L'architecture à multi niveaux

R ÉSUMÉ:

Dans tous les domaines notamment la gestion, l'être humain s'est mobilisé afin d'exploiter le temps comme il se doit.

Le bon fonctionnement d'une entreprise se base essentiellement sur une bonne organisation de ses services, éventuellement la gestion des stocks qui est l'élément crucial.

Dans notre cas, étant affectés à l'Entreprise Nationale des Industries en Electroménager (ENIEM), plus précisément, au sein de l'unité commerciale, le bon fonctionnement des mouvements des produits est d'une importance capitale dans la planification de la production ainsi que sur les ventes.

Notre projet consiste en la mise en place d'une application pour la comptabilisation des stocks des produits finis de l'ENIEM et porte sur l'implémentation d'une base de données et de développement d'une application client/serveur qui procurera un environnement convivial aux acteurs commerciaux et qui répondra aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Le présent rapport synthétise tout le travail que nous avons effectué dans cette perspective, il est organisé en cinq chapitres comme suit :

- § Le premier chapitre donne une présentation générale du projet : l'organisme d'accueil et les généralités sur les stocks.
- § Dans le second chapitre, nous procédons à un exposé de l'état de l'art du domaine qui nous concerne.
 - Nous présentons dans un premier temps le système existant pour dévoiler ses défaillances et ses limites.
 - Nous présentons également la solution que nous proposons afin de palier aux limites du système actuel.
- § Dans le troisième chapitre intitulé "Analyse et Spécification des besoins", nous présentons la démarche adoptée pour la modélisation de notre application en utilisant le langage de modélisation UML.
- § Dans le quatrième chapitre nous présentons la conception générale et la conception détaillée du système.
- § Le dernier chapitre décrit les taches accomplies dans la réalisation.

Enfin nous donnons une conclusion récapitulant le travail réalisé sans oublier de citer les références bibliographiques sur lesquelles nous nous sommes appuyées pour mener à bien ce travail.