

**BURKINA FASO  
UNITE-PROGRES-JUSTICE**

**MINISTERE DES ENSEIGNEMENTS SECONDAIRE,  
SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE**

-----  
**UNIVERSITE POLYTECHNIQUE DE BOBO-DIOULASSO**

-----  
**ECOLE SUPERIEURE D'INFORMATIQUE**



**MEMOIRE DE FIN DE CYCLE**

en vue de l'obtention du

**DIPLOME D'INGENIEUR DE CONCEPTION EN INFORMATIQUE**

**THEME :**

**Analyse Conception et Réalisation d'un outil de  
gestion budgétaire**

Présenté par :

**ISSOUF ABDOUL KADER TRAORE**

**Maître de stage : Mr Mamadou SOURABIE**

**Directeur de mémoire : Mr Sadouanouan MALO**

**N° : .....-2011/CICI3**

**JUILLET 2011**



**DEDICACES**

*A mes parents et particulièrement à ma mère*

*A mes amis*

## **REMERCIEMENTS**

*Au terme de ce travail, je tiens à remercier*

*Mon maître de stage, Mr Mamadou SOURABIE*

*Mon superviseur, Mr Sadouanouan MALO*

*Tout le personnel de la DAF*

*Tout le personnel de la direction de l'ESI*

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

<b>SOMMAIRE</b> .....	1
<b>SYGLES ET ABREVIATIONS</b> .....	3
<b>LISTE DES FIGURES</b> .....	4
<b>INTRODUCTION GENERALE</b> .....	6
<b>CHAPITRE 1 : ETUDE PREALABLE</b> .....	7
1. NOTE DE LANCEMENT.....	7
1.1 PRESENTATION DE LA STRUCTURE D'ACCUEIL.....	7
1.2 PROBLEMATIQUE ET OBJECTIFS.....	8
1.3 GESTION DU PROJET.....	9
1.4 PRESENTATION DE LA DEMARCHE SUIVIE.....	11
2. ETUDE DE L'EXISTANT.....	14
2.1 COMPTES RENDUS D'INTERVIEWS.....	14
2.2 PRESENTATION DE L'EXISTANT MATERIEL ET LOGICIEL.....	14
2.3 ANALYSES DE L'EXISTANT.....	14
2.4 DIAGNOSTICS DE L'EXISTANT.....	15
3. ETUDE DES BESOINS.....	16
3.1 MODELISATION DU CONTEXTE.....	16
3.2 SPECIFICATION DES BESOINS FONCTIONNELS.....	16
3.3 SPECIFICATION DES BESOINS TECHNIQUES.....	18
3.4 SCENARIO DU FUTUR SYSTEME.....	18
3.5 ESTIMATION DES COUTS.....	23
<b>CHAPITRE 2 : CONCEPTION DU FUTUR SYSTEME</b> .....	26
1. PHASE D'ELABORATION.....	26
1.1 DECOUPAGE EN CATEGORIES.....	26
1.2 MODELISATION STATIQUE.....	30
1.3 MODELISATION DYNAMIQUE.....	33
1.3.1 DIAGRRAMES DE SEQUENCES.....	33
1.3.2 DIAGRAMMES D'ACTIVITES.....	38
1.4 MODELE DE DEPLOIEMENT.....	44
2. POLITIQUE DE TRANSITOIRE.....	45
3. POLITIQUE DE SECURITE.....	47
3.1 PROTECTION CONTRE LES CATASTROPHES.....	47
3.2 PROTECTION CONTRE LES VIRUS INFORMATIQUES.....	48

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

3.3	PROTECTION CONTRE LES PANNES D'ELECTRICITE.....	48
3.4	CONFIDENTIALITE DES DONNEES.....	48
3.5	PROTECTION DES DONNEES.....	48
4.	PROCEDURES DE SECOURS.....	49
4.1	PANNE DE SERVEUR.....	49
4.2	POSTE DE TRAVAIL INDISPONIBLE.....	49
<b>CHAPITRE 3 : REALISATION.....</b>		<b>50</b>
1.	OUTILS DE MODELISATION.....	50
2.	LES OUTILS DE DEVELOPPEMENT.....	51
3.	QUELQUES CAPTURES D'ECRAN.....	54
4.	BILAN DU STAGE.....	66
4.1	PLANNING REEL.....	66
4.2	ANALYSE DES ECARTS.....	67
<b>CONCLUSION GENERALE.....</b>		<b>68</b>
<b>BIBLIOGRAPHIE ET WEBOGRAPHIE.....</b>		<b>69</b>
<b>UTILISATION DE LA BIBLIOGRAPHIE ET WEBOGRAPHIE.....</b>		<b>70</b>
<b>ANNEXE UML2.....</b>		<b>71</b>

## SIGLES ET ABREVIATIONS

Sigle ou abréviation	Signification
AC	Agent comptable
CICI	Cycle des Ingénieurs de Conception en Informatique
COCOMO	Constructive Cost Model
CSAF	Chef de service administratif et financier
DAF	Direction de l'administration et des finances
DAF	Directeur de l'administration et des finances
DCMEF	Directeur de contrôle des marchés public et des engagements financiers
DRH	Directeur des ressources humaines
EDI	Environnement de Développement Intégré
ESI	Ecole Supérieure d'Informatique
PCA	Président du conseil d'administration
PRM	Personne responsables des marchés
RA	Régie d'Avance
RAD	Rapid Application Development
S/DAF	Secrétariat de la DAF
SAF.C	Service administratif et financier central
SDP	Service du patrimoine
SDS	Service de la solde
SQL	Structured Query Language
UML	Unified Modeling Language
UPB	Université Polytechnique de Bobo-Dioulasso

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## LISTE DES FIGURES

FIGURES	PAGES
Figure 1 : Organigramme de la DAF de l'UPB	08
Figure 2 : Modèle du cycle vie en V	11
Figure 3 : Diagramme de contexte dynamique	16
Figure 4 : Décomposition du cas d'utilisation « Gérer profile d'utilisateur »	18
Figure 5 : Paquetage de diagrammes de cas d'utilisation « Administrer »	26
Figure 6 : Paquetage de diagrammes de cas d'utilisation « Gestion budget »	27
Figure 7 : Paquetage de cas d'utilisation « Gestion des achats»	28
Figure 8 : Paquetage de cas d'utilisation « autre dépense »	29
Figure 9 : Paquetages de Diagramme de classes « Administrer»	30
Figure 10 : Paquetages de Diagramme de classes « Gestion budget»	30
Figure 11 : Paquetages de Diagramme de classes « Gestion Autre dépense»	31
Figure 12 : Paquetages de Diagramme de classes « Gestion des achats»	32
Figure 13 : Diagramme de séquences du cas d'utilisation « Authentification »	33
Figure 14 : Diagramme de séquences du cas d'utilisation « Ajouter profile d'utilisateur »	34
Figure 15 : Diagramme de séquences du cas d'utilisation « Lister profile d'utilisateur »	35
Figure 16 : Diagramme de séquences du cas d'utilisation « Modifier profile d'utilisateur »	36
Figure 17 : Diagramme de séquences du cas d'utilisation « Supprimer profile d'utilisateur »	37
Figure 18 : Diagramme de séquences du cas d'utilisation « Approbation ordonnateur»	38
Figure 19 : Diagramme d'activités du cas d'utilisation « Authentification »	39
Figure 20 : Diagramme d'activités du cas d'utilisation « Ajouter profile d'utilisateur »	40
Figure 21 : Diagramme d'activités du cas d'utilisation « Lister profil d'utilisateur »	41
Figure 22 : Diagramme d'activités du cas d'utilisation « Modifier profil d'utilisateur »	42
Figure 23 : Diagramme d'activités du cas d'utilisation « Supprimer profil d'utilisateur»	43
Figure 24 : Diagramme d'activités du cas d'utilisation « Approbation ordonnateur»	44
Figure 25 : Schéma du modèle de déploiement	45
Figure 26 : Aperçu de l'interface de TIAK IDE	52
Figure 27 : Aperçu de la page d'accueil	53
Figure 28 : Aperçu de la page d'édition de chapitres	53
Figure 29 : Aperçu de la page d'édition de comptes	54
Figure 30 : Aperçu de la page d'édition de titres	55
Figure 31 : Aperçu de la page d'édition de demande de prix	56
Figure 32 : Aperçu de la page d'édition de proposition de prix	57
Figure 33 : Aperçu de la page d'édition de contrats	58
Figure 34 : Aperçu de la page d'engagements	59
Figure 35 : Aperçu de la page d'édition de liquidations	60
Figure 36 : Aperçu de la page d'émission de mandats	61
Figure 37 : Aperçu de la page d'édition de mandats de paiements	62

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

Figure 38 : Aperçu de la page d'édition de comptes d'utilisateurs	63
Figure 39 : Aperçu de la page de gestion de mot de passe	63
Figure 40 : Aperçu de la page d'édition de sauvegarde de données	64
Figure 41 : Aperçu de la page d'édition de restaurations de données	64



# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## INTRODUCTION GENERALE

L'Ecole Supérieure d'Informatique (ESI) a été créée en 1991, elle est actuellement rattachée à l'Université Polytechnique de Bobo-Dioulasso (UPB). L'ESI a pour mission de former des étudiants aux diplômes suivants : l'ingéniorat de travaux informatiques niveau (BAC+3) dans les options analyse et programmation (AP), réseaux et maintenance (REMI) à travers le Cycle d'Ingénieurs de Travaux Informatiques (CITI); l'ingéniorat de conception informatique à travers le Cycle des Ingénieurs de Conception en Informatique (CICI) niveau (BAC+5); le Diplôme d'Etude Approfondies (DEA) à travers un troisième cycle.

Dans le but de joindre la pratique à la théorie, les étudiants en troisième année du Cycle d'Ingénieurs de Conception en Informatique, sont appelés à effectuer un stage pratique dans les institutions ou entreprises. C'est dans ce cadre que la Direction de l'Administration et des Finances (DAF) de l'UPB nous a ouvert ses portes pour un stage pratique sous le thème : **Analyse Conception et Réalisation d'un outil de gestion budgétaire.**

## CHAPITRE 1 : ETUDE PREALABLE

Dans cette partie nous allons présenter l'organisation générale de la DAF de l'UPB ensuite le problème de la gestion actuelle du fonctionnement. Puis nous présenterons par la suite l'étude de l'existant et l'étude des besoins des utilisateurs.

### **1. NOTE DE LANCEMENT**

#### **1.1 PRESENTATION DE LA STRUCTURE D'ACCUEIL**

##### **• PRESENTATION DE L'UPB**

L'Université Polytechnique de Bobo-Dioulasso est un Etablissement Public de l'Etat à Caractère Scientifique, Culturel et Technique (EPSCT), chargé d'enseignement supérieur et de recherche scientifique. Elle est située dans le village de Nasso à 15km de Bobo-Dioulasso qui est la capitale économique du Burkina Faso.

Elle a pour mission l'élaboration et la transmission de la connaissance pour la formation des hommes et des femmes afin de répondre aux besoins de la Nation.

Pour parvenir à sa mission l'UPB s'est fixée les objectifs suivants :

- former des cadres dans tous les domaines en général et dans les filières professionnalisantes en particulier ;
- conduire des activités de recherche scientifique et en vulgariser les résultats ;
- élever le niveau technique, scientifique et culturel des étudiants pour une ouverture sur le marché de l'emploi et les secteurs de production ;
- délivrer des titres et diplômes ;
- valoriser les compétences dans tous les secteurs d'activité du pays.

##### **• PRESENTATION DE LA DIRECTION DE L'ADMINISTRATION ET DES FINANCES (DAF)**

La Direction de l'Administration et des Finances(DAF) est chargée des opérations administratives et financières de l'UPB. Elle centralise tous les renseignements concernant les finances et le patrimoine de l'UPB et en assure la gestion conformément aux règles de la comptabilité publique en vigueur au Burkina Faso.

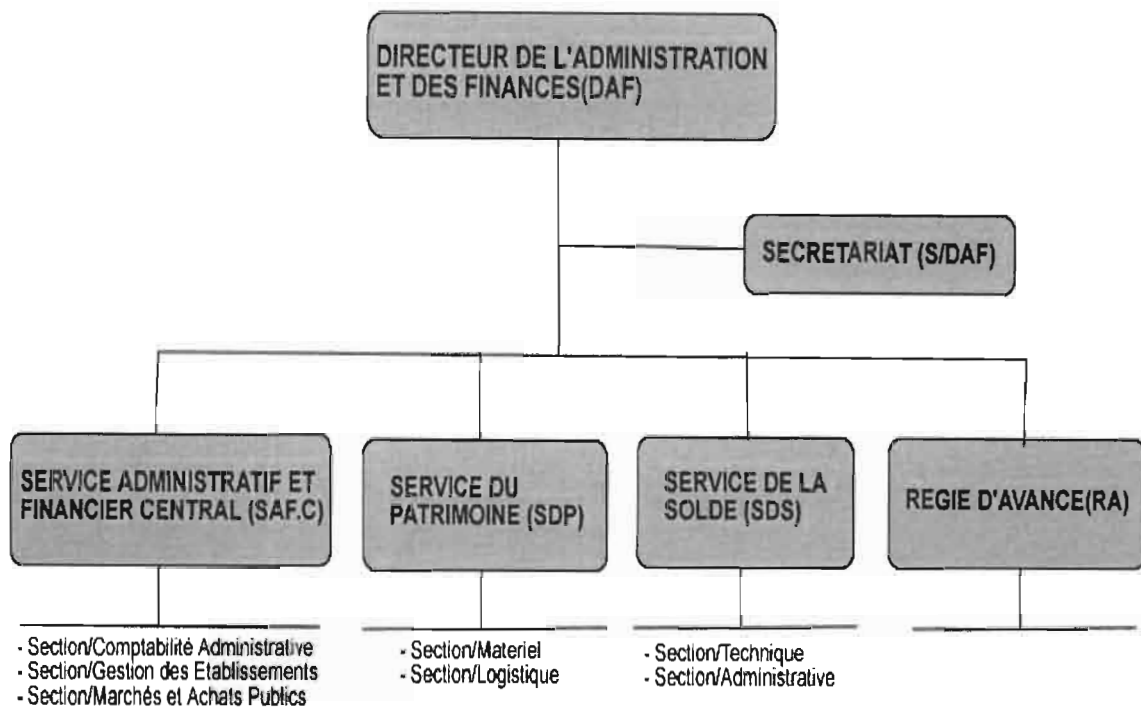


Figure 1 : Organigramme de la DAF de l'UPB

## 1.2 PROBLEMATIQUE ET OBJECTIFS

### • PROBLEMATIQUE

La DAF de l'UPB est dotée d'un logiciel de traitement des salaires ; cependant elle ne possède aucun logiciel de gestion du budget qui est l'élément central de la DAF. Ainsi donc les documents tels que les bons de commandes, les bordereaux, les mandats sont établis manuellement et passent successivement au niveau des différentes sections pour traitement jusqu'à la fin du circuit où les dossiers sont archivés pour en garder les traces. De même le budget est manuellement établi et garde ses traces sur papier.

Nous avons ainsi pas mal de problèmes liés au système actuel :

- La lenteur des traitements du au temps de passage physique des dossiers au niveau du circuit de traitement ;
- Les risques de perte de dossiers au cours du circuit ;
- Des difficultés d'archivage de dossiers et de recherche de dossiers archivés ;

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

## • OBJECTIFS DU PROJET

Le projet d'étude vise à élaborer un outil de gestion budgétaire permettant de résoudre les problèmes du système actuel tout en permettant des possibilités d'évolution. Ainsi donc l'outil de gestion budgétaire devra offrir les fonctionnalités suivantes :

- La gestion des lignes budgétaires
  - ✓ Création et modification des exercices, chapitres, articles, paragraphes et comptes
  - ✓ Le suivi des imputations et l'évolution des crédits
- La gestion des Titres
  - ✓ Enregistrement et modification des titres
- La gestion des sections
  - ✓ Enregistrement et modification sections
- Gestion des dépenses
  - ✓ Enregistrement, modification des différentes dépenses
  - ✓ Enregistrement, modification des informations liées aux dépenses pour chaque étape du traitement
  - ✓ La génération et impression de documents tels que les bons, mandats et bordereaux, les fiches d'engagement.

## 1.3 GESTION DU PROJET

La gestion de projet (ou conduite de projet) est une démarche visant à organiser de bout en bout le bon déroulement d'un projet.

### • CYCLE DE VIE D'UN LOGICIEL

Le cycle de vie d'un logiciel (en anglais software life cycle), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

Le cycle de vie du logiciel comprend généralement au minimum les étapes suivantes :

- **Définition des objectifs:**

Cette étape consiste à définir la finalité du projet et son inscription dans une stratégie globale ;

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **Analyse des besoins et faisabilité:**

C'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins ;

- **Spécifications ou conception générale:**

Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel ;

- **Conception détaillée:**

Cette étape consiste à définir précisément chaque sous-ensemble du logiciel ;

- **Codage (Implémentation ou programmation):**

C'est la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception ;

- **Tests unitaires:**

Ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications ;

- **Intégration :**

L'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document ;

- **Qualification (ou recette) :**

C'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales ;

- **Documentation :**

Elle vise à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs ;

- **Mise en production :**

C'est le déploiement sur site du logiciel ;

- **Maintenance :**

Elle comprend toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement. Le cycle de vie permet de prendre en compte, en plus des aspects techniques, l'organisation et les aspects humains. Ils existent plusieurs modèles de cycles de vie d'un logiciel tels que : Modèle en cascade, en V, en spirale, par incrément, etc.

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## • LE CYCLE DE VIE EN V

Le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Il s'agit d'un modèle en cascade dans lequel le développement des tests et des logiciels sont effectués de manière synchrone. Il retient certaines étapes du cycle de vie en général, ces étapes vous sont présentés sur le schéma de la figure 2 ci-dessous.

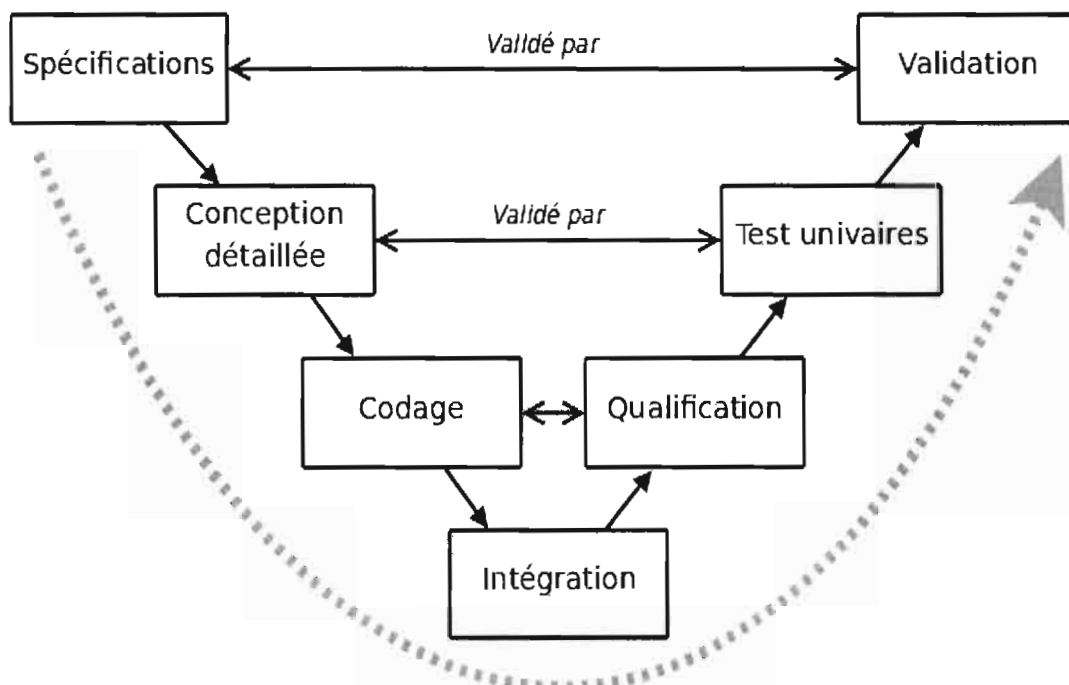


Figure 2 : Modèle du cycle vie en V

Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. En particulier il permet de commencer plus tôt :  
Le plan de tests de qualification, le plan d'évaluation des performances. Cependant, ce modèle souffre toujours du problème de la vérification trop tardive du bon fonctionnement du système.

## 1.4 PRESENTATION DE LA DEMARCHE SUIVIE

### • LES ACTEURS DU PROJET

On peut subdiviser les acteurs du projet en trois groupes :

#### ○ LE GROUPE DE PILOTAGE

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

Il a pour rôle de prendre les décisions relatives aux objectifs visés. Il fixe les orientations générales, les délais et détermine les moyens à mettre en place pour la réalisation du projet. Il est composé de :

- ✓ **Mr. Mamadou SOURABIE, maitre de stage**
- ✓ **Mr. Sadouanouan MALO, directeur de mémoire**

## ○ **LE GROUPE DE PROJET**

Il est chargé de recenser les besoins des utilisateurs, de les formaliser puis de produire l'application. Il est composé de :

- ✓ **Mr. Issouf Abdoul Kader TRAORE,**

## ○ **LE GROUPE DES UTILISATEURS**

Fournit les informations nécessaires à la bonne marche du développement de l'application. Il est composé de : **tout le personnel de la DAF.**

## • **PLANING PREVISIONNEL**

Le planning prévisionnel de notre travail se résume dans le tableau ci-dessous :

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

Phases	Activités	Période	Durée
Lancement		02 au 08 Novembre 2011	7 jours
Spécification	Capture des besoins fonctionnels Analyse Capture des besoins techniques Analyse et choix de la solution de mise en œuvre Modélisation du système futur	09 Novembre 2011 Au 10 Décembre 2011	32 jours
Conception détaillée	Conception détaillée des classes Conception des couches logicielles	11 Décembre au 30 Décembre 2011	20 jours
Réalisation et mise en œuvre	Codage et test Recette	31 Décembre 2011 au 02 Février 2012	2 mois

### • MOYENS UTILISES

Nous avons utilisés les moyens suivants :

- Ordinateur portable HP Compaq nc6400 : Core Duo 1.66 GH X 1.66 GH de processeur, 1Go de RAM, avec Windows 7 comme système d'exploitation ;
- Ordinateur de bureau HP d530 : 2 .4 GH de processeur, 1Go de RAM, avec Windows XP professionnel SP3 comme système d'exploitation
- Une clé de connexion internet 3G de 7.2 MB/S
- Logiciels
  - ✓ Visual studio 2010
  - ✓ Oracle XE
  - ✓ TIAK IDE
  - ✓ ORACLE DATA PROVIDER FOR.NET
  - ✓ VISUAL PARADIGM



# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

## 2. ETUDE DE L'EXISTANT

Ce point nous permet de faire le compte rendu des interviews de présenter l'existant et d'en faire un diagnostic.

### 2.1 COMPTES RENDUS D'INTERVIEWS

Les personnes que nous avons rencontrées en vue d'inventorier les besoins principaux, de recenser les besoins fonctionnels et d'appréhender les besoins non fonctionnels sont :

- M. Den Boni Agent de la DAF UPB
- M. Ouattara Agent du service comptable UPB
- M. Dondassé Agent de la DAF UPB
- M. Sourabié Agent de la DAF UPB
- Mme Traoré Agent de la DAF UPB
- Le Agents de la chaine solde UPB
- Les agents du contrôle financier UPB
- M. Ouedraogo Agent de la logistique UPB

### 2.2 PRESENTATION DE L'EXISTANT MATERIEL ET LOGICIEL

#### • PRESENTATION DE L'EXISTANT LOGICIEL

Aucun logiciel de gestion du budget n'existe au sein de la DAF de l'UPB.

#### • PRESENTATION DE L'EXISTANT MATERIEL

Tous les services de la DAF sont dotés d'ordinateurs de bureau pour la plus part pentium 4 ou dual core. Aucun réseau informatique n'est dédié à la DAF.

### 2.3 ANALYSES DE L'EXISTANT

#### • IDENTIFICATION DES ACTEURS

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. L'analyse textuelle du cahier de charge révèle les acteurs suivants :

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

- L'ordonnateur
- Le Directeur de l'Administration de l'administration et des finances (DAF)
- Le chef de structure
- Le chef de section dépense
- Le chef de section recette
- Le Directeur de Contrôle des Marchés Public et des Engagements Financiers(DCMEF)
- Le chef de chaine solde
- Chef comptable
- Le Chef de Service Administratif et Financier (CSAF)
- Le Président du Conseil d'Administration (PCA)
- Personne Responsable des Marchés(PRM)

## • GESTION ACTUELLE DU SYSTEME

La gestion actuelle du budget de l'UPB est entièrement manuelle, il n'y a aucune application permettant de gérer le budget.

## 2.4 DIAGNOSTICS DE L'EXISTANT

Nous avons ainsi pas mal de problèmes liés au système actuel :

- La lenteur des traitements dus au temps de passage physique des dossiers au niveau du circuit de traitement ;
- Les risques de perte de dossiers au cours du circuit ;
- Des difficultés d'archivage de dossiers et de recherche de dossiers archivés ;

## 3. ETUDE DES BESOINS

### 3.1 MODELISATION DU CONTEXTE

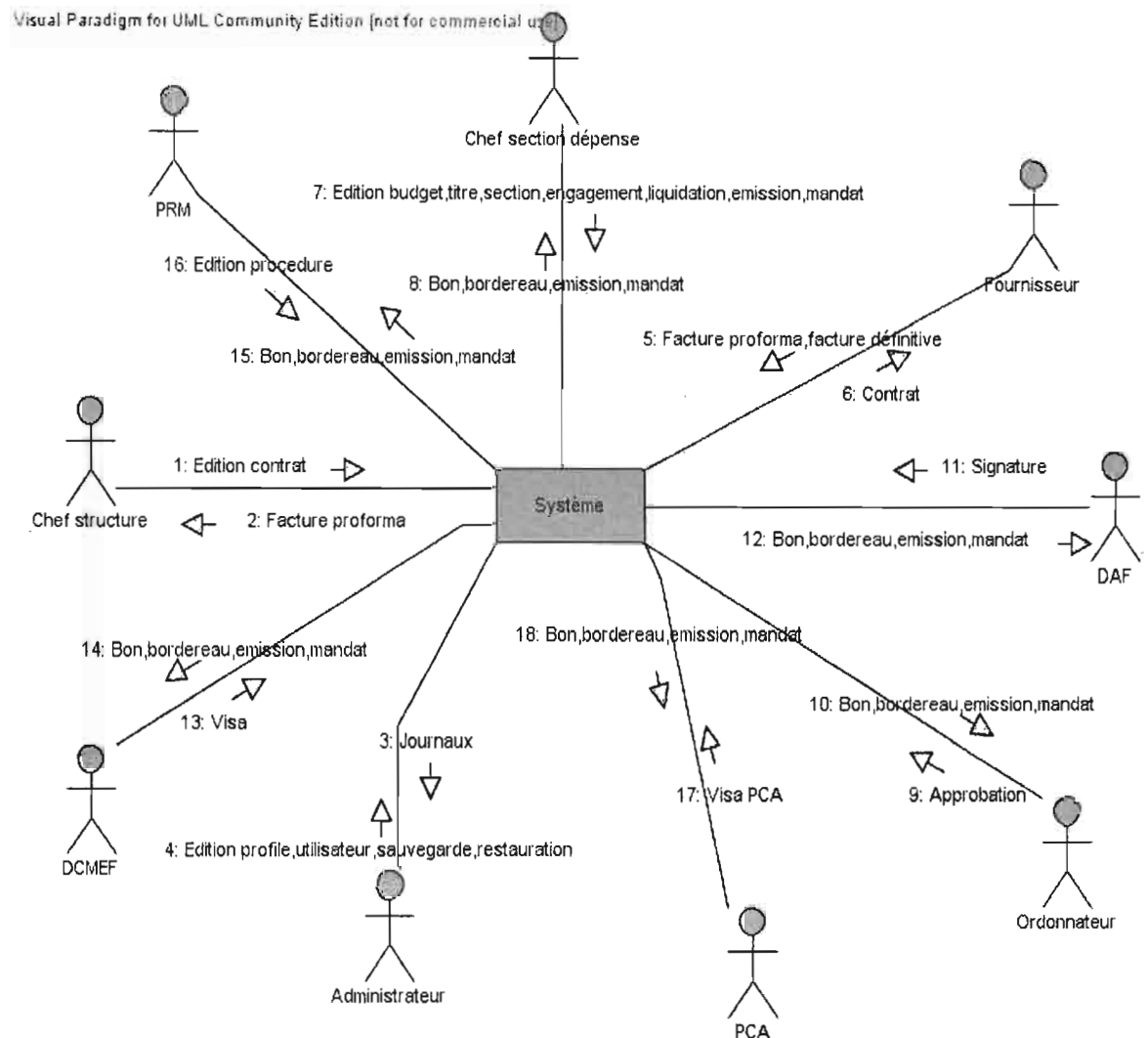


Figure 3 : Diagramme de contexte dynamique

### 3.2 SPECIFICATION DES BESOINS FONCTIONNELS

Les différents acteurs ont eu à exprimer des besoins qui peuvent se résumer aux fonctionnalités suivantes que le futur système devra fournir :

- Gestion du budget
  - Créer, modifier un exercice
  - Créer, modifier un chapitre
  - Créer, modifier un article
  - Créer, modifier un paragraphe

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

- Créer, modifier un compte
- Créer, modifier un titre
- Créer, modifier une section
- Créer, modifier Réaménagement
- Créer, modifier Situation de recouvrement
- Gestion des dépenses
  - Créer, modifier une banque
  - Créer, modifier une dépense simple
  - Créer, modifier une procédure
  - Créer, modifier une facturation
  - Créer, modifier un contrat
  - Créer, modifier un engagement
  - Créer, modifier une livraison
  - Créer, modifier une facture définitive
  - Créer, modifier une liquidation
  - Créer, modifier une émission de mandat de paiement
  - Créer, modifier un de mandat de paiement
  - Créer, modifier un virement bancaire
  - Créer, modifier un PV de réception
  - Générer des documents imprimables tels que :
    - ✓ Contrats
    - ✓ Bordereaux d'émission de mandant
    - ✓ Bordereaux de réception
    - ✓ Bordereaux de liquidation
    - ✓ Fiches de contrôle de dépenses
    - ✓ Le mandat chronologique
    - ✓ La fiche des dépenses engagées
- Administration
  - Créer, modifier les profiles d'utilisateurs
  - Créer, modifier les comptes d'utilisateurs
  - Sauvegarder les données
  - Restaurer les données

## 3.3 SPECIFICATION DES BESOINS TECHNIQUES

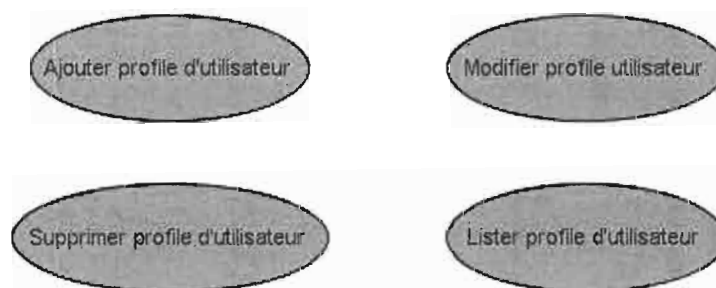
- L'application doit offrir une interface assez simple et conviviale aux utilisateurs
- L'utilisateur doit être guidé par des informations réactives renvoyés par l'application pour le guider au cours de l'utilisation
- L'application doit garantir la sécurité à travers les droits d'accès accordés à chaque profile
- L'application doit être une application réseaux permettant ainsi à plusieurs personnes d'accéder à la base de données en mêmes temps

## 3.4 SCENARIO DU FUTUR SYSTEME

La majorité des cas d'utilisations peuvent se subdiviser en quatre cas d'utilisations (Créer, modifier, lister et supprimer), seuls les cas d'utilisations « Authentification », « Visa », « Approbation » et « Signature » font exception.

Nous avons ainsi un exemple de décomposition du cas d'utilisation « Gérer profile d'utilisateur » ci-dessous :

Visual Paradigm for UML Community Edition [not for commercial use]



**Figure 4 : Décomposition du cas d'utilisation « Gérer profile d'utilisateur »**

Nous présenterons dans ce qui suit la description textuelle de tous les types cas d'utilisations possibles à travers la description textuelle des cas d'utilisations « Authentification », « Créer un profile d'utilisateur », « Modifier un profile d'utilisateur », « Lister un profile d'utilisateur », « Supprimer un profile d'utilisateur », « Approbation Ordonnateur » :

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- Description textuelle du cas d'utilisation « Authentification »

SOMMAIRE D'IDENTIFICATION	
TITRE	Authentification
BUT	Connexion à l'espace de travail
RESUME	L'utilisateur introduit son login et mot de passe pour accéder à l'espace de travail
ACTEUR	Tout utilisateur confondu
DESCRIPTION DES ENCHAINEMENTS	
PRE CONDITIONS	POST CONDITIONS
L'utilisateur doit avoir un compte d'utilisateur	Accès à son espace de travail
SCENARIO NOMINAL	
<ol style="list-style-type: none"><li>1. L'utilisateur demande le formulaire d'authentification</li><li>2. Le système affiche le formulaire d'authentification</li><li>3. L'utilisateur saisit son login, son mot de passe et valide</li><li>4. Le système vérifie les champs (A1)</li><li>5. Le système vérifie l'existence du compte d'utilisateur (A2)</li><li>6. Le système affiche l'espace de travail</li></ol>	
ENCHAINEMENT ALTERNATIF	
A1 : Champs obligatoires non valides et/ou vides <ol style="list-style-type: none"><li>1. Le système affiche un message d'erreur</li><li>2. Le scénario reprend en 3.</li></ol> A2 : Login ou mot de passe non valide <ol style="list-style-type: none"><li>1. Le système affiche un message d'erreur</li><li>2. Le scénario reprend en 3</li></ol>	

- Description textuelle du cas d'utilisation « Ajouter profile d'utilisateur »

SOMMAIRE D'IDENTIFICATION	
TITRE	Ajouter profile d'utilisateur
BUT	Créer un profile d'utilisateur
RESUME	L'administrateur saisie le nom du profile, sélectionne les droits d'accès puis enregistre
ACTEUR	Administrateur
DESCRIPTION DES ENCHAINEMENTS	

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

PRE CONDITIONS	POST CONDITIONS
L'administrateur est authentifié	Création d'un nouveau profile d'utilisateur
SCENARIO NOMINAL	
<ol style="list-style-type: none"> <li>1. L'utilisateur demande le formulaire d'édition de profiles</li> <li>2. Le système charge la liste des droits d'accès</li> <li>3. Le système affiche le formulaire d'édition de profiles avec la liste des droits d'accès</li> <li>4. L'administrateur saisit le nom du profile, sélectionne les droits d'accès souhaités puis clique sur enregistrer</li> <li>5. Le système vérifie les champs (A1)</li> <li>6. Le système vérifie l'existence du profile d'utilisateur (A2)</li> <li>7. Le système enregistre les informations</li> <li>8. Le système affiche un message de succès d'enregistrement</li> </ol>	
ENCHAINEMENT ALTERNATIF	
<p><b>A1</b> : Champs obligatoires non valides et/ou vides</p> <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le scénario reprend en 4.</li> </ol> <p><b>A2</b> : Le nom du profile d'utilisateur existant pas déjà</p> <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le scénario reprend en 4.</li> </ol>	

- Description textuelle du cas d'utilisation « Lister profile d'utilisateur »

SOMMAIRE D'IDENTIFICATION	
<b>TITRE</b>	Lister profile d'utilisateur
<b>BUT</b>	Obtenir la liste des profiles d'utilisateur
<b>RESUME</b>	L'administrateur demande la liste des profiles d'utilisateur que le système affiche éventuellement
<b>ACTEUR</b>	Administrateur
DESCRIPTION DES ENCHAINEMENTS	
PRE CONDITIONS	POST CONDITIONS
L'administrateur est authentifié	Affichage de la liste des profiles ou message de liste vide
SCENARIO NOMINAL	
<ol style="list-style-type: none"> <li>1. L'administrateur demande le formulaire de listage de profiles d'utilisateur</li> <li>2. Le système affiche le formulaire de listage de profiles d'utilisateur</li> <li>3. L'administrateur clique sur lister</li> </ol>	

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

4. Le système charge la liste de profils d'utilisateur (A1)
5. Le système affiche la liste des profils
<b>ENCHAÎNEMENT ALTERNATIF</b>
<b>A1</b> : La liste des profils d'utilisateur est vides
1. Le système affiche le message « Liste de profils d'utilisateur vide »,

- Description textuelle du cas d'utilisation « Modifier profile d'utilisateur »

<b>SOMMAIRE D'IDENTIFICATION</b>	
<b>TITRE</b>	Modifier profile d'utilisateur
<b>BUT</b>	Modifier un profile d'utilisateur existant
<b>RESUME</b>	L'administrateur sélectionne un profile d'utilisateur modifie les informations puis enregistre
<b>ACTEUR</b>	Administrateur
<b>DESCRIPTION DES ENCHAÎNEMENTS</b>	
<b>PRE CONDITIONS</b>	<b>POST CONDITIONS</b>
L'administrateur est authentifié et le formulaire de listage des profils est affiché avec la liste des profils d'utilisateur	Modification du profile d'utilisateur concerné
<b>SCENARIO NOMINAL</b>	
<ol style="list-style-type: none"> <li>1. L'administrateur sélectionne un profile d'utilisateur</li> <li>2. Le système affiche les informations du profile d'utilisateur en mode modification</li> <li>3. L'administrateur complète ou corrige les informations puis clique sur enregistrer</li> <li>4. Le système vérifie les champs (A1)</li> <li>5. Le système vérifie l'existence du profile d'utilisateur (A2)</li> <li>6. Le système enregistre les informations</li> <li>7. Le système affiche un message de succès d'enregistrement</li> </ol>	
<b>ENCHAÎNEMENT ALTERNATIF</b>	
<b>A1</b> : Champs obligatoires non valides et/ou vides	
<ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le scénario reprend en 3.</li> </ol>	
<b>A2</b> : Le nom du profile d'utilisateur existant déjà	
<ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le scénario reprend en 3.</li> </ol>	

- Description textuelle du cas d'utilisation « Supprimer profile d'utilisateur »



## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

SOMMAIRE D'IDENTIFICATION	
TITRE	Supprimer profile d'utilisateur
BUT	Supprimer un profile d'utilisateur existant
RESUME	L'administrateur sélectionne un profile d'utilisateur puis le supprime
ACTEUR	Administrateur
DESCRIPTION DES ENCHAINEMENTS	
PRE CONDITIONS	POST CONDITIONS
L'administrateur est authentifié et le formulaire de listage des profiles est affiché avec la liste des profiles d'utilisateur	Suppression du profile d'utilisateur concerné
SCENARIO NOMINAL	
<ol style="list-style-type: none"> <li>1. L'administrateur sélectionne un profile d'utilisateur</li> <li>2. L'administrateur clique sur supprimer</li> <li>3. Le système vérifie l'existence du profile d'utilisateur (A1)</li> <li>4. Le système supprime le profile</li> <li>5. Le système affiche un message de succès de suppression</li> </ol>	
ENCHAINEMENT ALTERNATIF	
<b>A1</b> : Champs obligatoires non valides et/ou vides <ol style="list-style-type: none"> <li>1. Le système affiche un message « Afficher profile déjà modifié ou supprimé »</li> </ol>	

- Description textuelle du cas d'utilisation « Supprimer profile d'utilisateur »

SOMMAIRE D'IDENTIFICATION	
TITRE	Approbation ordonnateur
BUT	Activer ou désactiver l'approbation de l'ordonnateur
RESUME	L'ordonnateur demande la liste des éléments requis le système affiche éventuellement l'Ordonnateur active ou désactive l'option Approbation
ACTEUR	Ordonnateur
DESCRIPTION DES ENCHAINEMENTS	
PRE CONDITIONS	POST CONDITIONS
L'Ordonnateur est authentifié et le formulaire de listage des éléments est affiché avec la liste	Activation ou désactivation de l'approbation
SCENARIO NOMINAL	
<ol style="list-style-type: none"> <li>1. L'Ordonnateur sélectionne un élément</li> </ol>	

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

2. L'Ordonnateur clique pour cocher ou décocher puis clique sur enregistrer
3. Le système vérifie les champs (A1)
4. Le système vérifie l'existence de l'élément (A2)
5. Le système enregistre les informations
6. Le système affiche un message de succès d'enregistrement

## ENCHAÎNEMENT ALTERNATIF

**A1** : Champs obligatoires non valides et/ou vides

1. Le système affiche un message d'erreur
2. Le scénario reprend en 2.

**A2** : Le nom du profile d'utilisateur existant déjà

1. Le système affiche un message d'erreur
2. Le scénario reprend en 2.

### 3.5 ESTIMATION DES COUTS

Constructive Cost Model (COCOMO) est un modèle développé par Boehm en 1981 permettant d'estimer la taille du logiciel à développer et d'en déduire certaines contraintes telles que les délais, le nombre de personne nécessaire, le coût.

#### • Principes de la méthode COCOMO

Type de projet	Charge en mois homme	Durée en mois
Simple	$C = 3,2 (Kisl)^{1,05}$	$D = 2,5(C)^{0,38}$
Moyen	$C = 3 (Kisl)^{1,12}$	$D = 2,5(C)^{0,35}$
Complexe	$C = 2,8 (Kisl)^{1,2}$	$D = 2,5(C)^{0,32}$

C : charger en mois hommes

Soit l la taille du logiciel, on a :

- Projet simple si  $l < 50$  Kisl, spécifications stables, petite équipe.
- Projet moyen si  $300 > l > 50$  Kisl, spécifications stables, petite équipe.
- Projet complexe si  $l > 300$  Kisl, grande équipe.

#### • Application de la méthode COCOMO à notre projet

Dans notre cas le nombre de lignes de code source est estimé à 20 000 lignes de code sources, le projet est donc de type simple, soit :

- $l = 20$  Kisl ;
- $C = 3,2(20)^{1,05} = 74,34$  H/M ;
- $D = 2,5(C)^{0,38} = 2,5(74,34)^{0,38} = 12,85$  mois soit 12 mois 26 jours ;

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

- Taille moyenne de l'équipe :  $T = 74,34 / 12,85 = 5,78$  soit 6 personnes ;
- Coût du projet :  $\text{Cout} = C * 200\ 000 = 74,34 * 200\ 000 = 14\ 868\ 000$  FCFA.

- **Calcul du coût de formation du personnel**

On a 12 personnes au minimum à former conformément aux rôles décrits, ce qui nous donne le tableau suivant :

Prix horaire	Nombre d'heures	Nombre d'utilisateurs	Prix total
3000	10	12	360 000
<b>Coût total</b>			<b>360 000</b>

- **Coût de déploiement**

Nous estimons qu'il n'y a aucun matériel ni logiciel à acheter, car les équipements et les systèmes d'exploitation déjà présents au sein de la DAF suffisent à mettre en marche notre nouveau système moyennant une réadaptation de certains équipements à d'autres fins. Ceci étant nous proposons un coût forfaitaire de 150 000 F pour le déploiement.

- **Coût de maintenance**

Etant donné qu'il est assez difficile de prévoir quelle type de maintenance serait nécessaire, nous proposons également un coût forfaitaire de **150 000** au module maintenu de manière évolutive ou corrective au delà des délais de la phase d'insertion en entreprise convenu avec la DAF.

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **Tableaux récapitulatifs des coûts et délais du développement**

<b>INTITULE</b>	<b>VALEURS ET UNITES</b>
Délais	12 mois 26 jours
Taille moyenne de l'équipe	6 personnes
Coût de Développement	14 868 000 FCFA
Coût de Formation	360 000 FCFA
Coût des outils logiciels utilisés	0 FCFA
Coût de déploiement	150 000
<b>Coût Total de développement</b>	<b>15 378 000 FCFA</b>
Coût de maintenance	150 000 F au module

## CHAPITRE 2 : CONCEPTION DU FUTUR SYSTEME

Dans cette partie allons concevoir le futur système en regroupant les différentes parties en catégories, nous présenterons les politiques transitoires et les politiques de secours.

### 1. PHASE D'ELABORATION

#### 1.1 DECOUPAGE EN CATEGORIES

Nous avons ainsi effectué une subdivision de nos cas d'utilisations en quatre paquetages qui sont les suivants : « Gestion budget », « Gestion autre dépense », « Gestion des achats » et « Administrer ».

Visual Paradigm for UML Community Edition [not for commercial use]

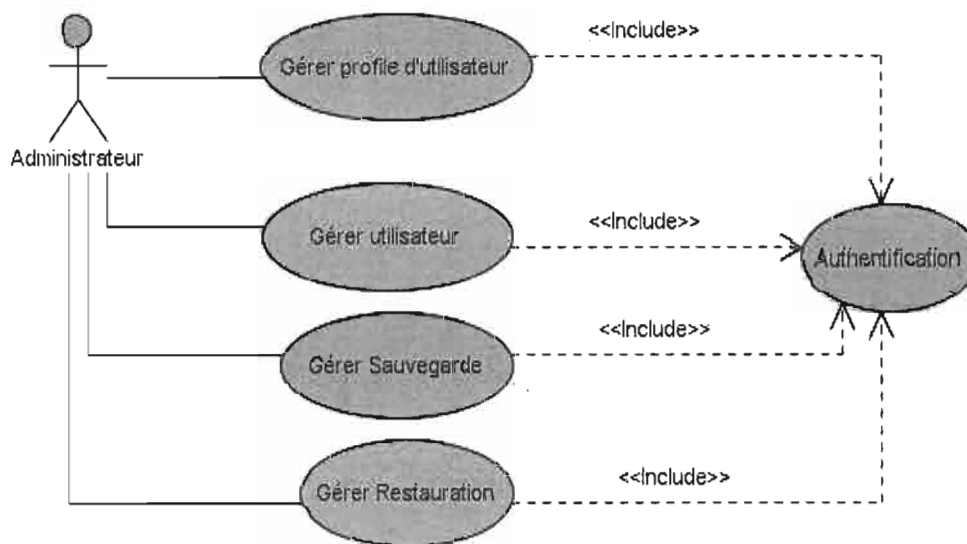


Figure 5 : Paquetage de diagrammes de cas d'utilisation « Administrer »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

Visual Paradigm for UML Community Edition (not for commercial use)

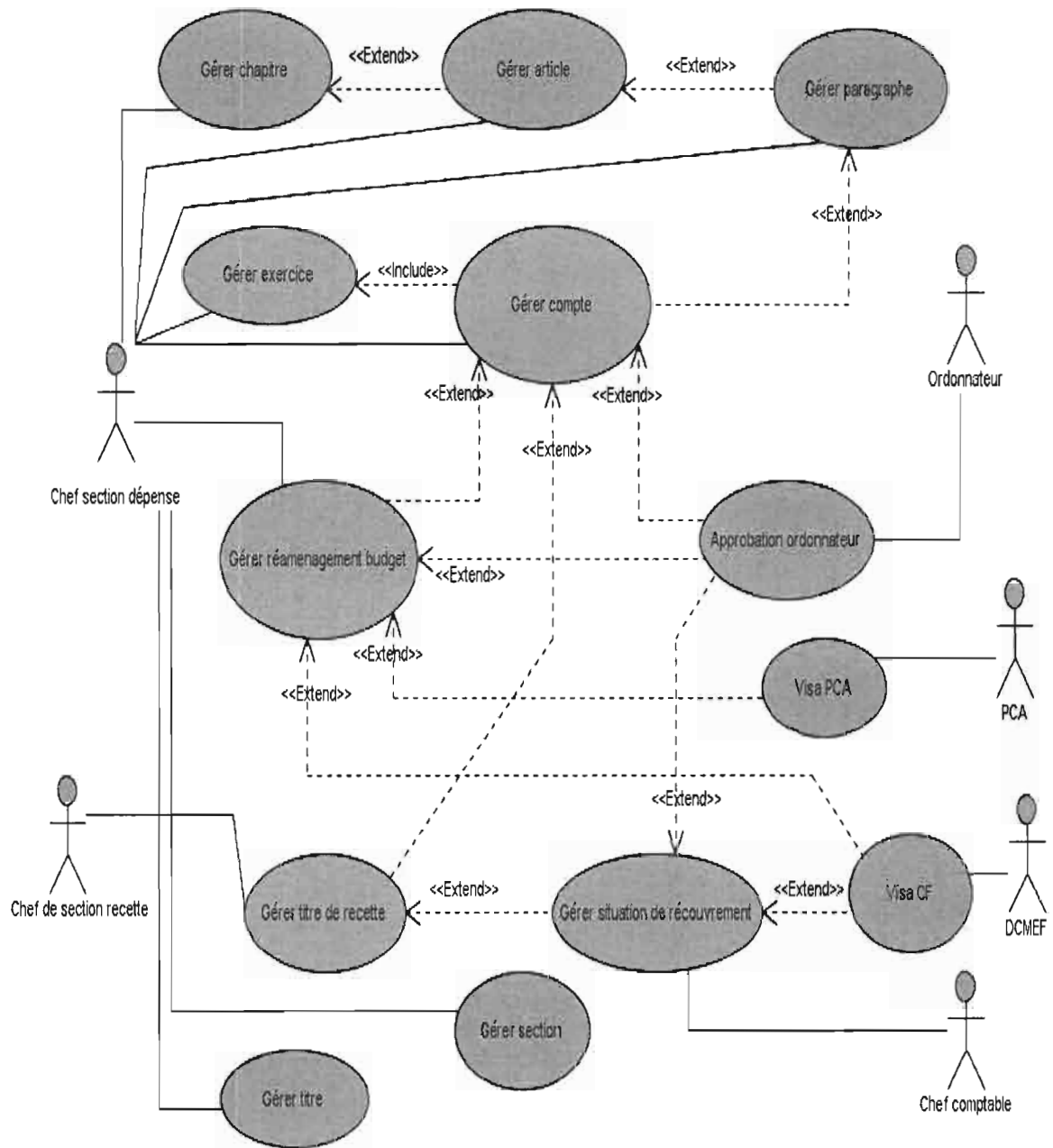


Figure 6 : Paquetage de diagrammes de cas d'utilisation « Gestion budget »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

Visual Paradigm for UML Community Edition [not for commercial use]

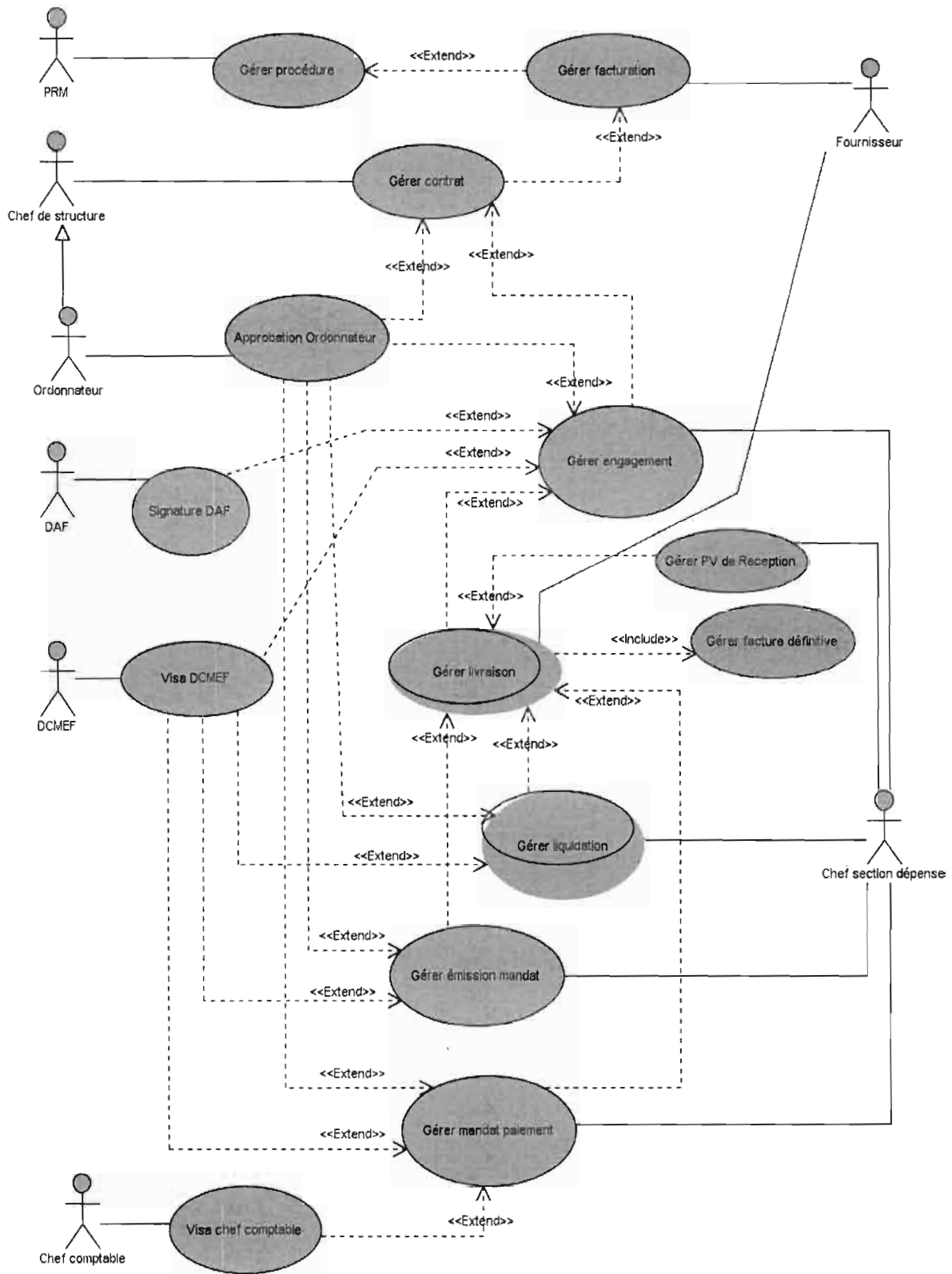


Figure 7 : Paquetage de cas d'utilisation « Gestion des achats »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

Visual Paradigm for UML Community Edition [not for commercial use]

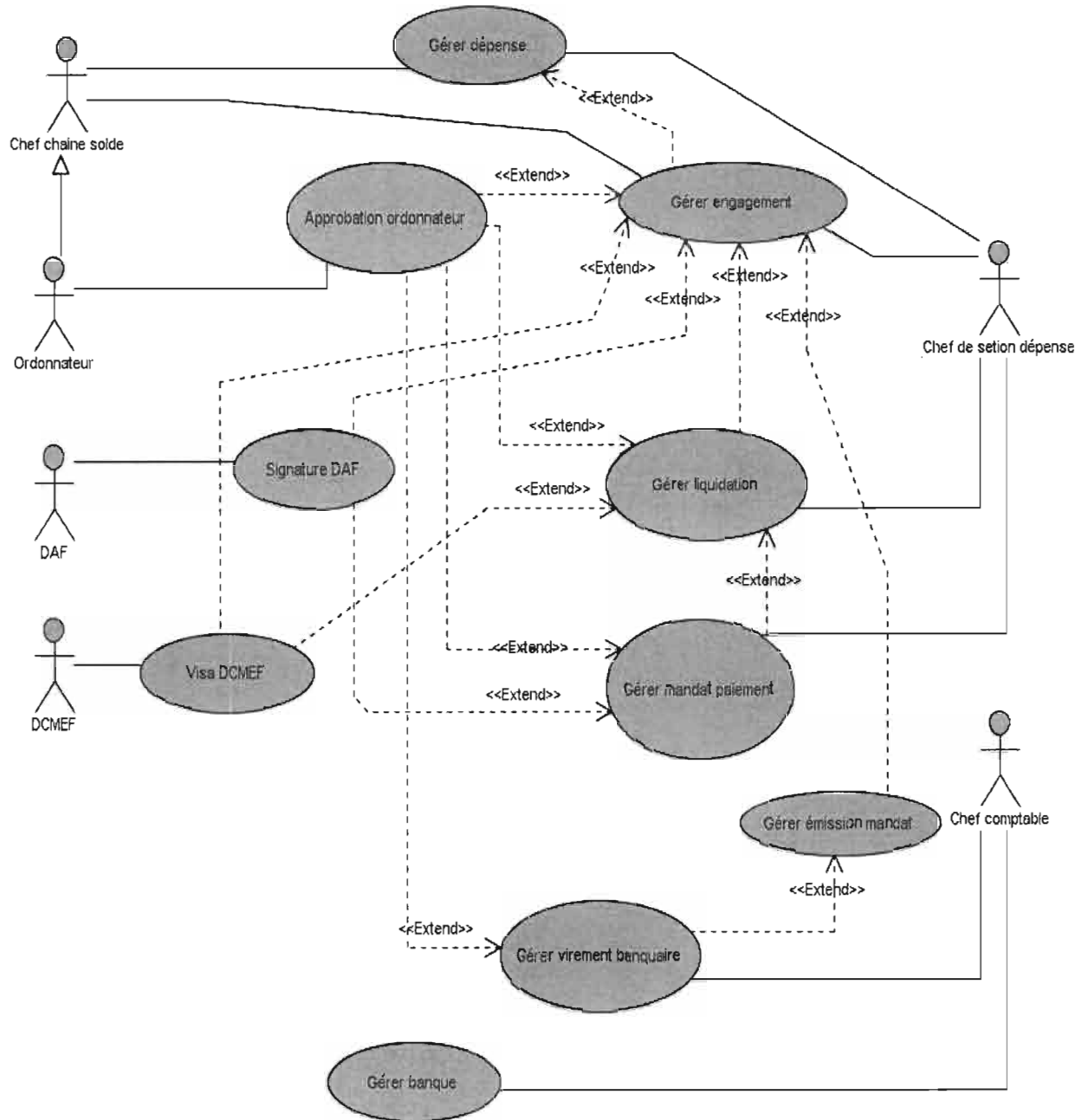


Figure 8 : Paquetage de cas d'utilisation « autre dépense »



## 1.2 MODELISATION STATIQUE

### • DIAGRAMMES DE CLASSES « Administrer »

Visual Paradigm for UML Community Edition [not for commercial use]

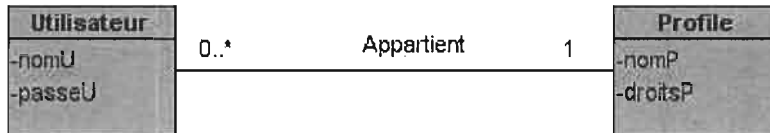


Figure 9 : Paquetages de Diagramme de classes « Administrer»

### • DIAGRAMMES DE CLASSES « Gestion budget »

Visual Paradigm for UML Community Edition [not for commercial use]

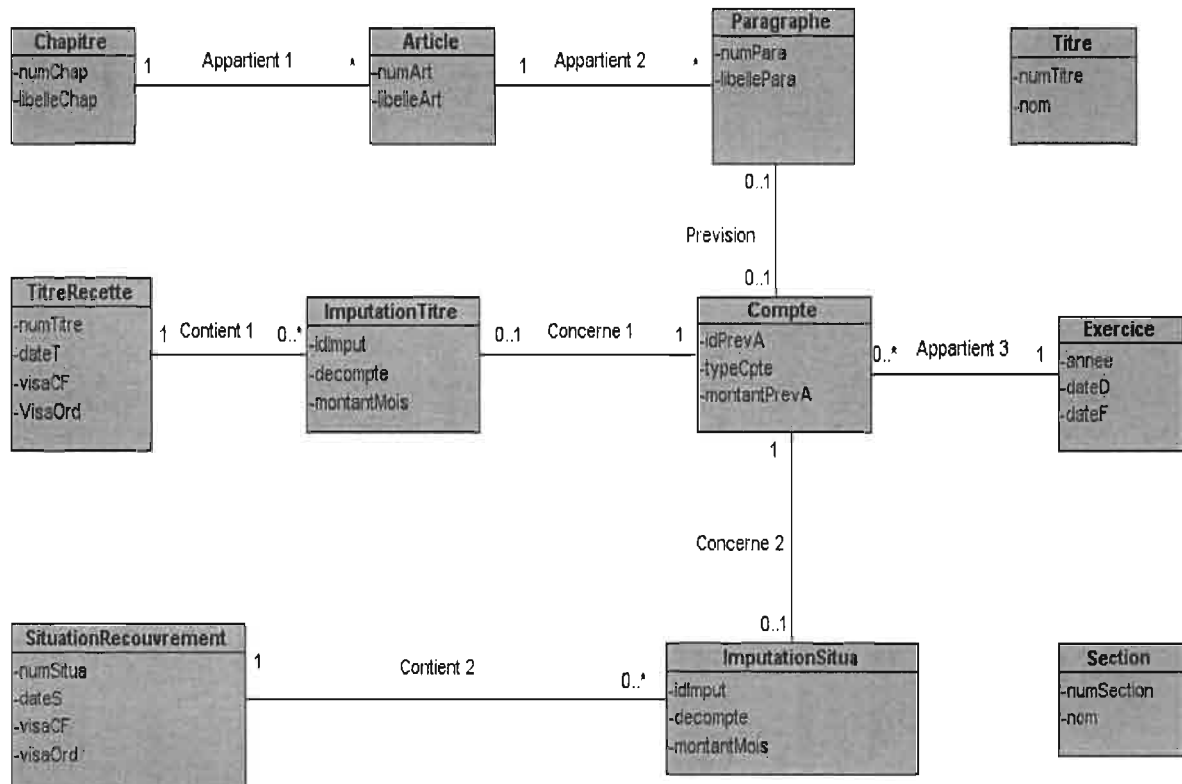


Figure 10 : Paquetages de Diagramme de classes « Gestion budget»

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## • DIAGRAMMES DE CLASSES « Gestion Autre dépense »

Visual Paradigm for UML Community Edition [not for commercial use]

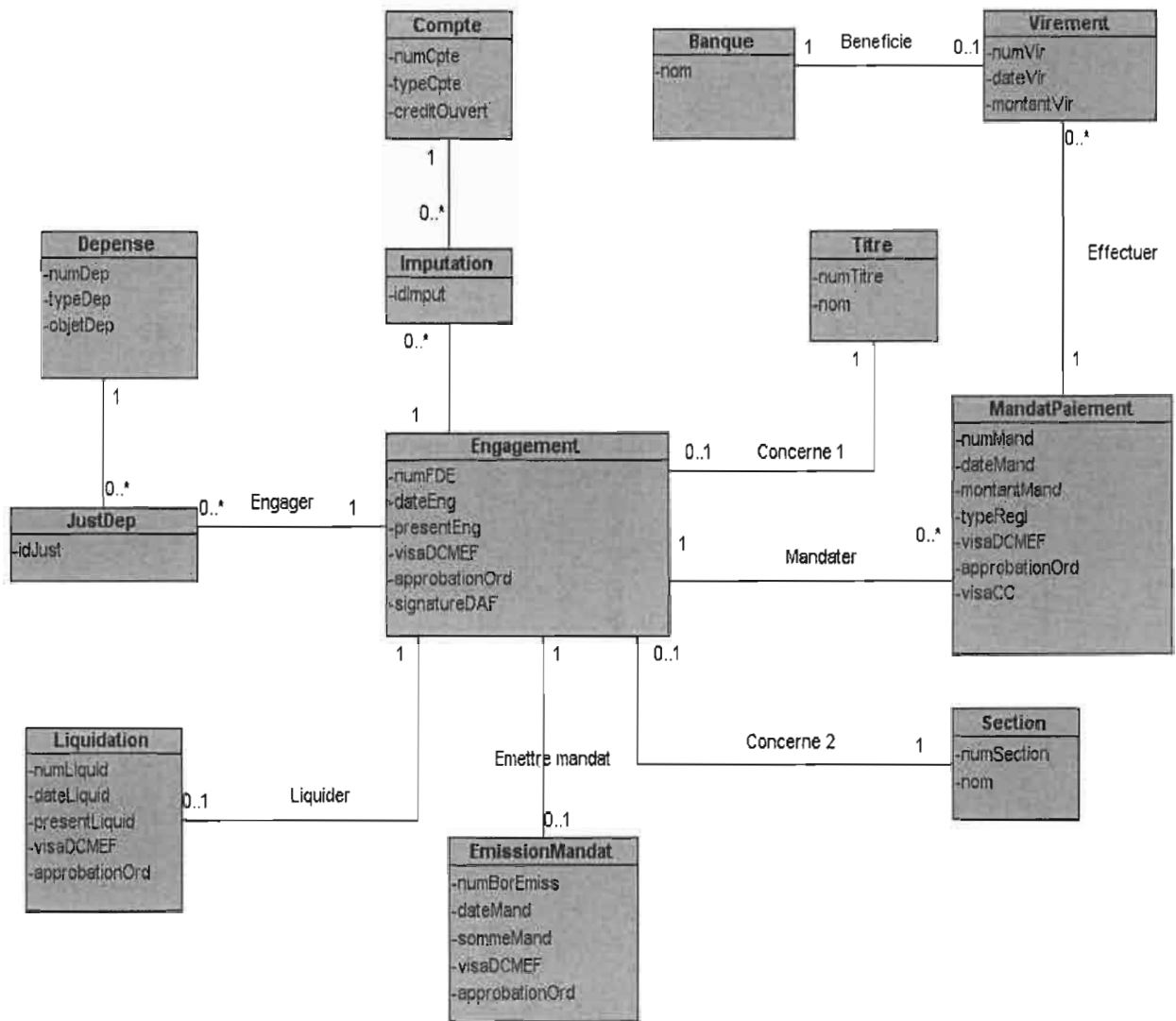


Figure 11 : Paquetages de Diagramme de classes « Gestion Autre dépense »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## • DIAGRAMMES DE CLASSES « Gestion des achats »

Visual Paradigm for UML Community Edition [not for commercial use]

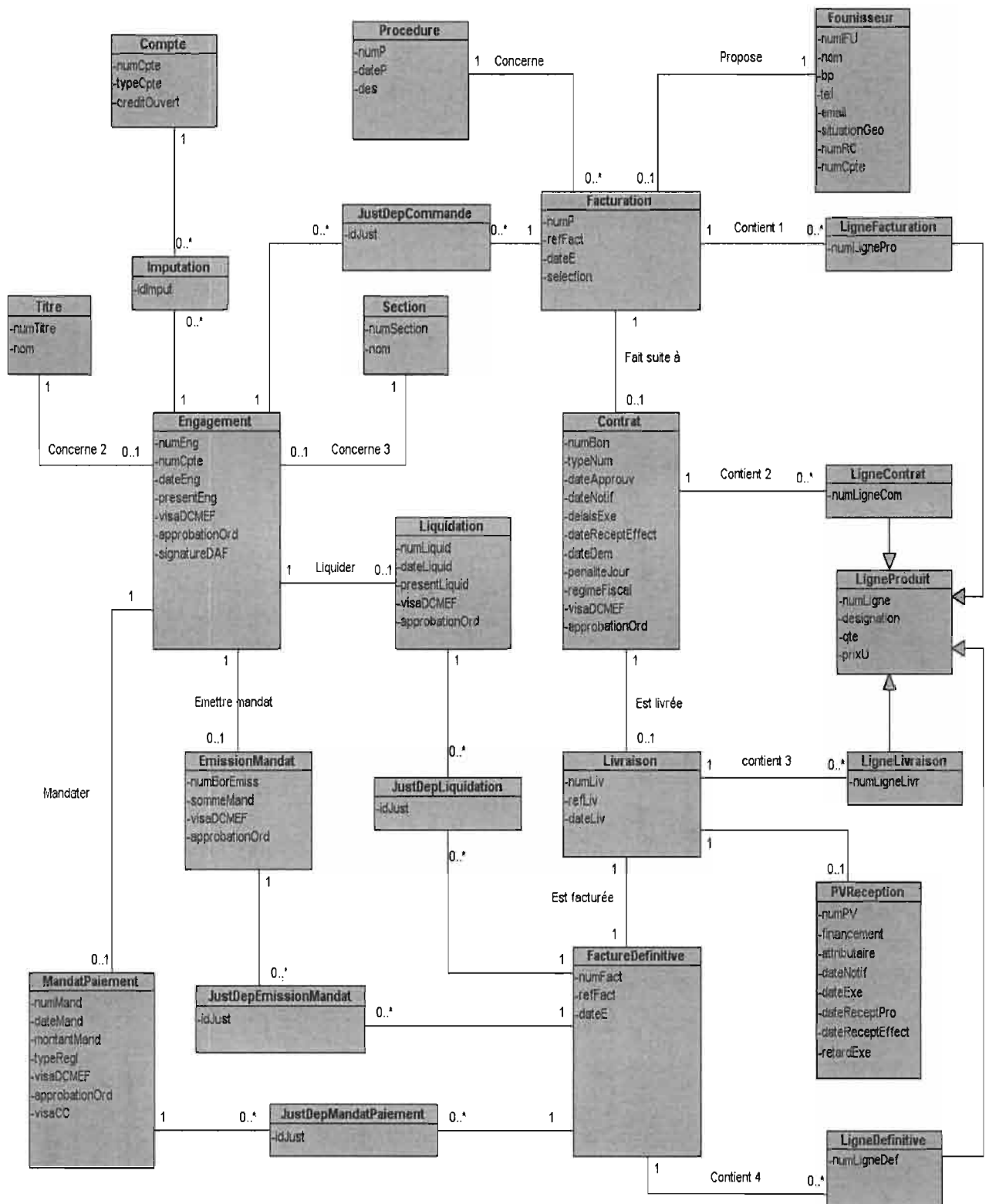


Figure 12 : Paquetages de Diagramme de classes « Gestion des achats »

## 1.3 MODELISATION DYNAMIQUE

### 1.3.1 DIAGRAMMES DE SEQUENCES

- **DIAGRAMMES DE SEQUENCES DU CAS D'UTILISATION**

- **« Authentification »**

Visual Paradigm for UML Community Edition [not for commercial use]

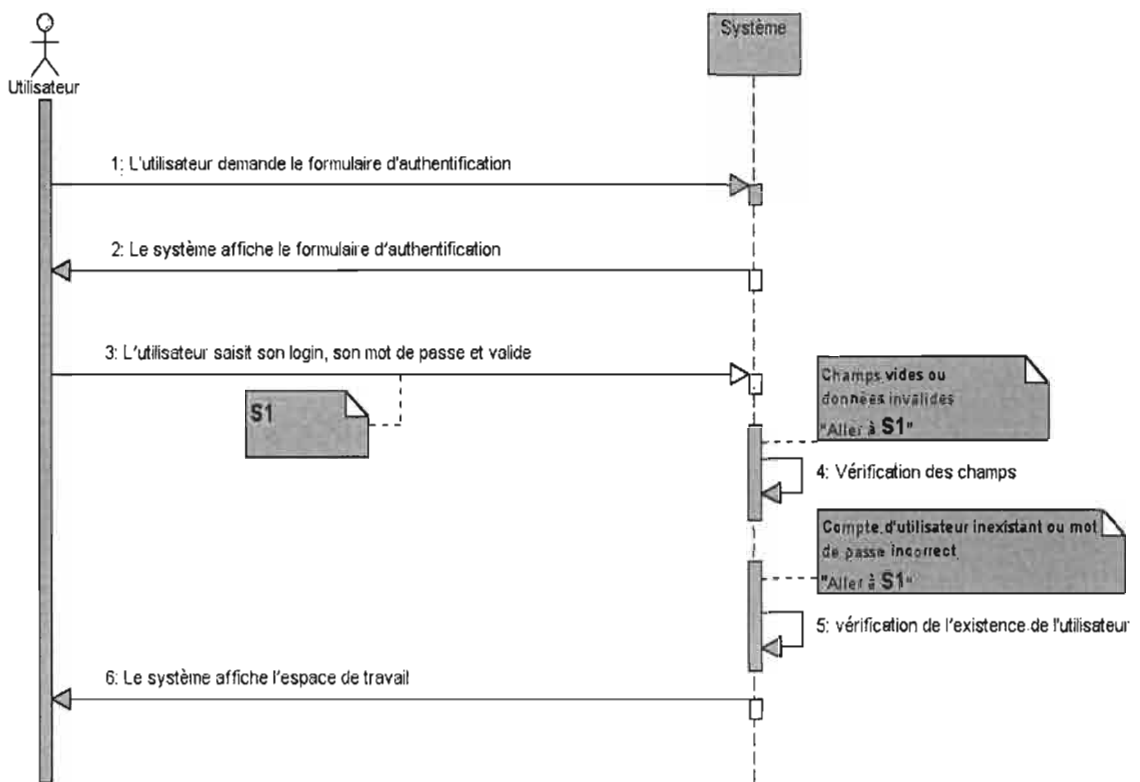
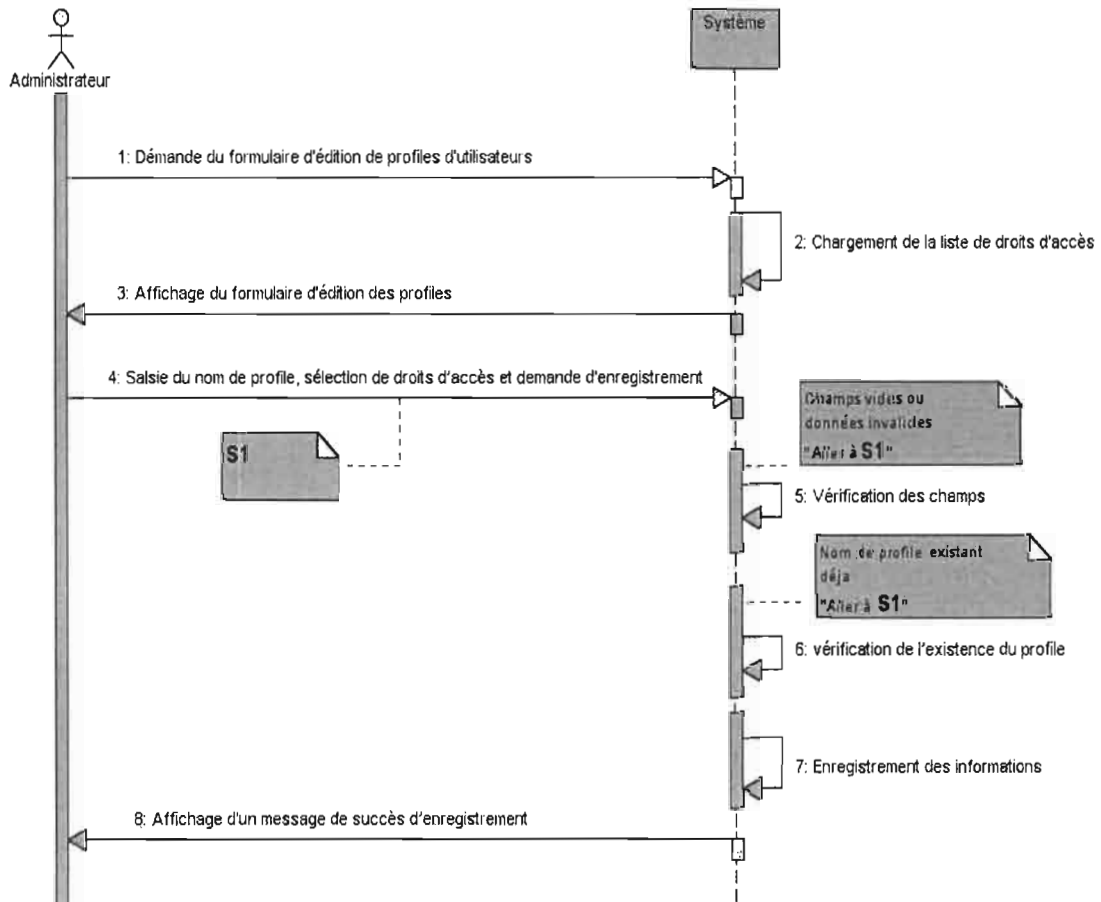


Figure 13 : Diagramme de séquences du cas d'utilisation « Authentification »

- **DIAGRAMMES DE SEQUENCES DU CAS D'UTILISATION « Ajouter profile d'utilisateur »**

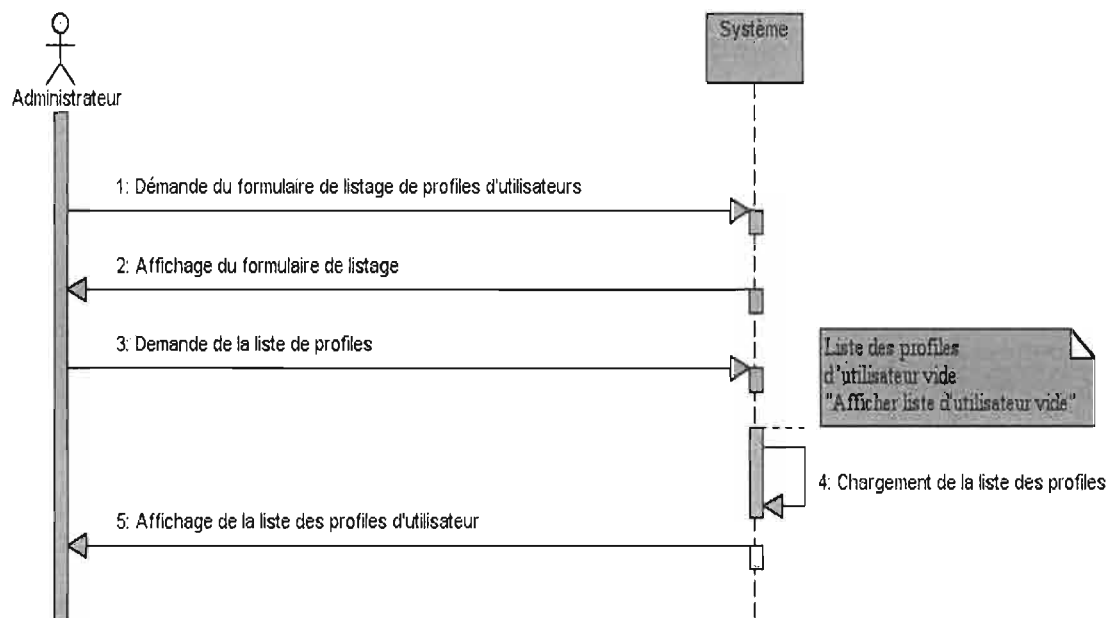
Visual Paradigm for UML Community Edition [not for commercial use]



**Figure 14 : Diagramme de séquences du cas d'utilisation « Ajouter profile d'utilisateur »**

- **DIAGRAMMES DE SEQUENCES DU CAS D'UTILISATION « Lister profile d'utilisateur »**

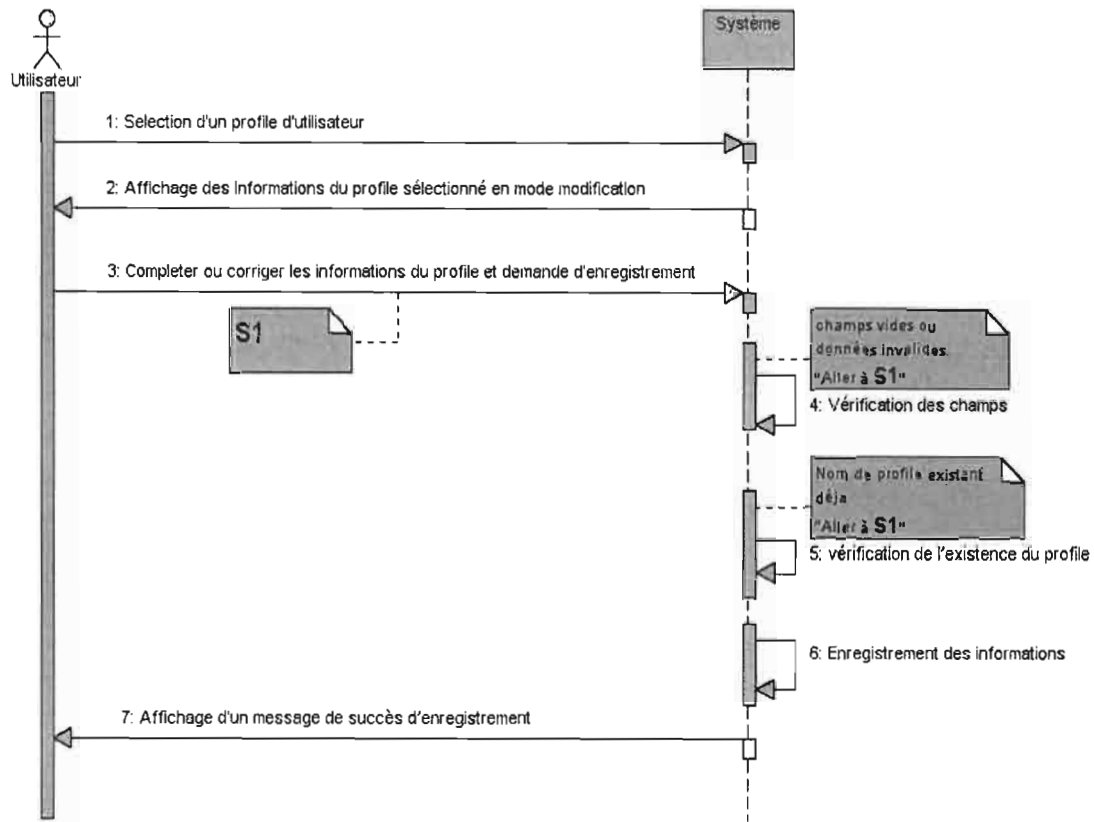
Visual Paradigm for UML Community Edition [not for commercial use]



**Figure 15 : Diagramme de séquences du cas d'utilisation « Lister profile d'utilisateur »**

- **DIAGRAMMES DE SEQUENCES DU CAS D'UTILISATION « Modifier profile d'utilisateur »**

Visual Paradigm for UML Community Edition (not for commercial use)



**Figure 16 : Diagramme de séquences du cas d'utilisation « Modifier profile d'utilisateur »**

- DIAGRAMMES DE SEQUENCES DU CAS D'UTILISATION « Supprimer profile d'utilisateur »

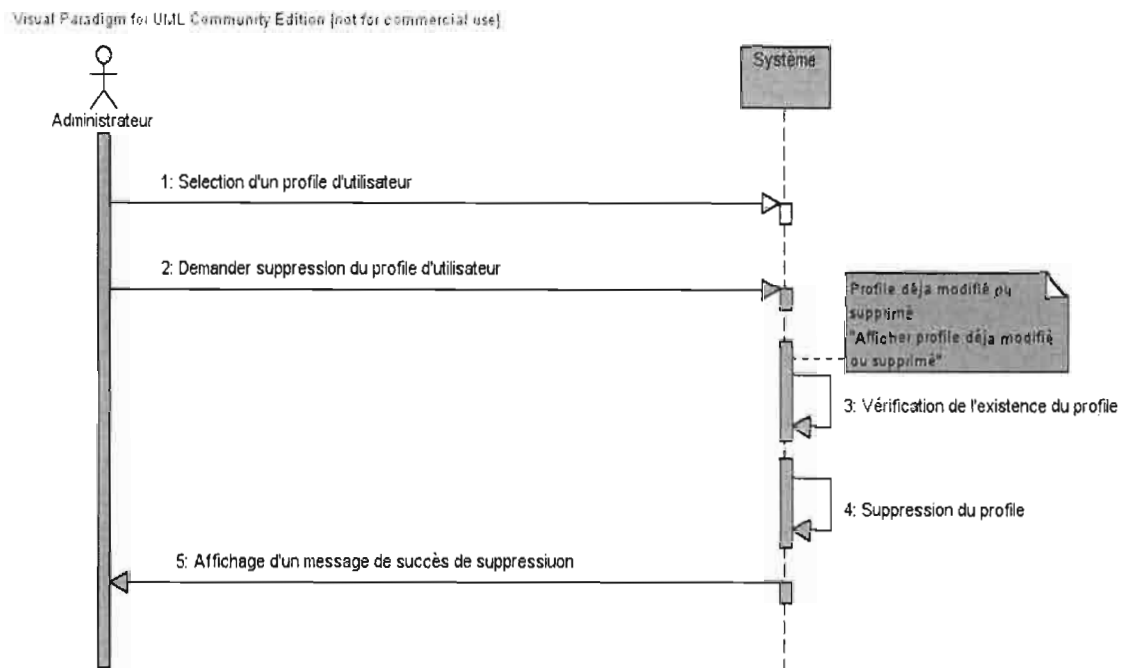
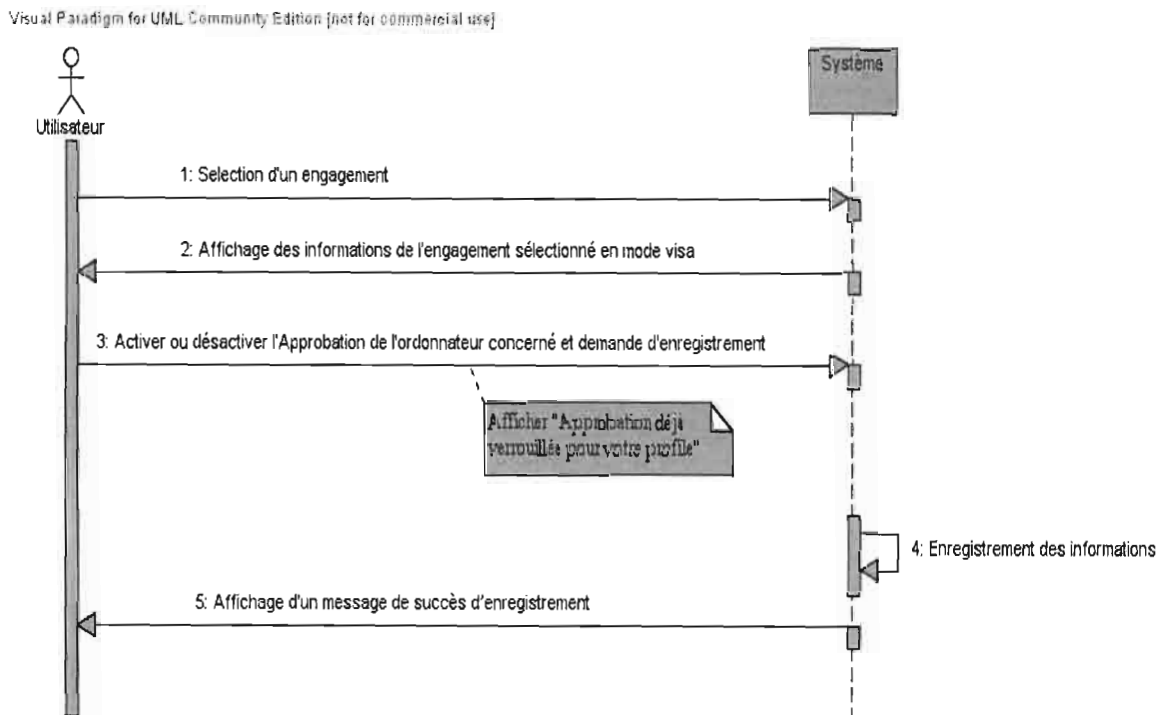


Figure 17 : Diagramme de séquences du cas d'utilisation « Supprimer profile d'utilisateur »



- **DIAGRAMMES DE SEQUENCES DU CAS D'UTILISATION « Approbation Ordonnateur »**



**Figure 18 : Diagramme de séquences du cas d'utilisation « Approbation ordonnateur»**

### 1.3.2 DIAGRAMMES DE D'ACTIVITES

Le diagramme d'activités est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activités correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Dans ce qui suit, nous présentons les diagrammes d'activités pour quelques cas d'utilisation dans notre système.

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## • DIAGRAMMES D'ACTIVITES DU CAS D'UTILISATION « Authentification »

Visual Paradigm for UML Community Edition [not for commercial use]

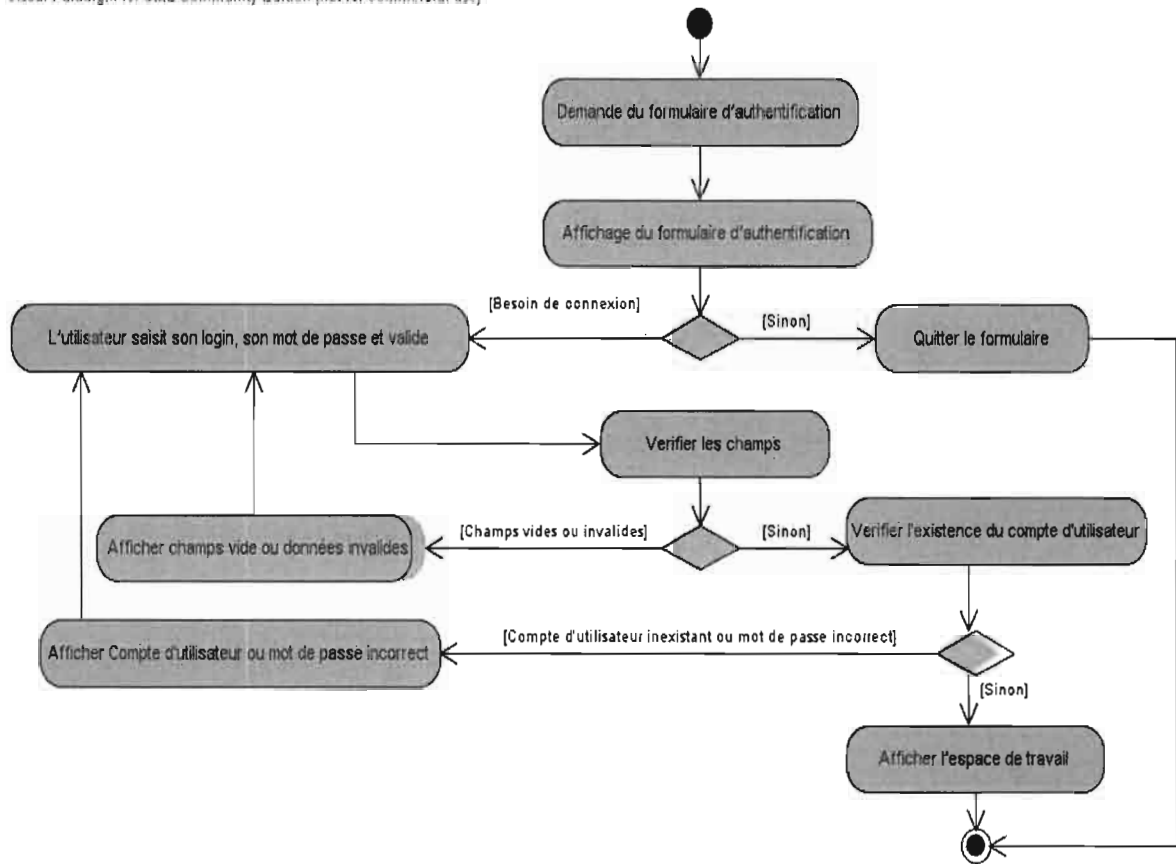


Figure 19 : Diagramme d'activités du cas d'utilisation « Authentification »

- **DIAGRAMMES D'ACTIVITES DU CAS D'UTILISATION « Ajouter profile d'utilisateur »**

Visual Paradigm for UML Community Edition [not for commercial use]

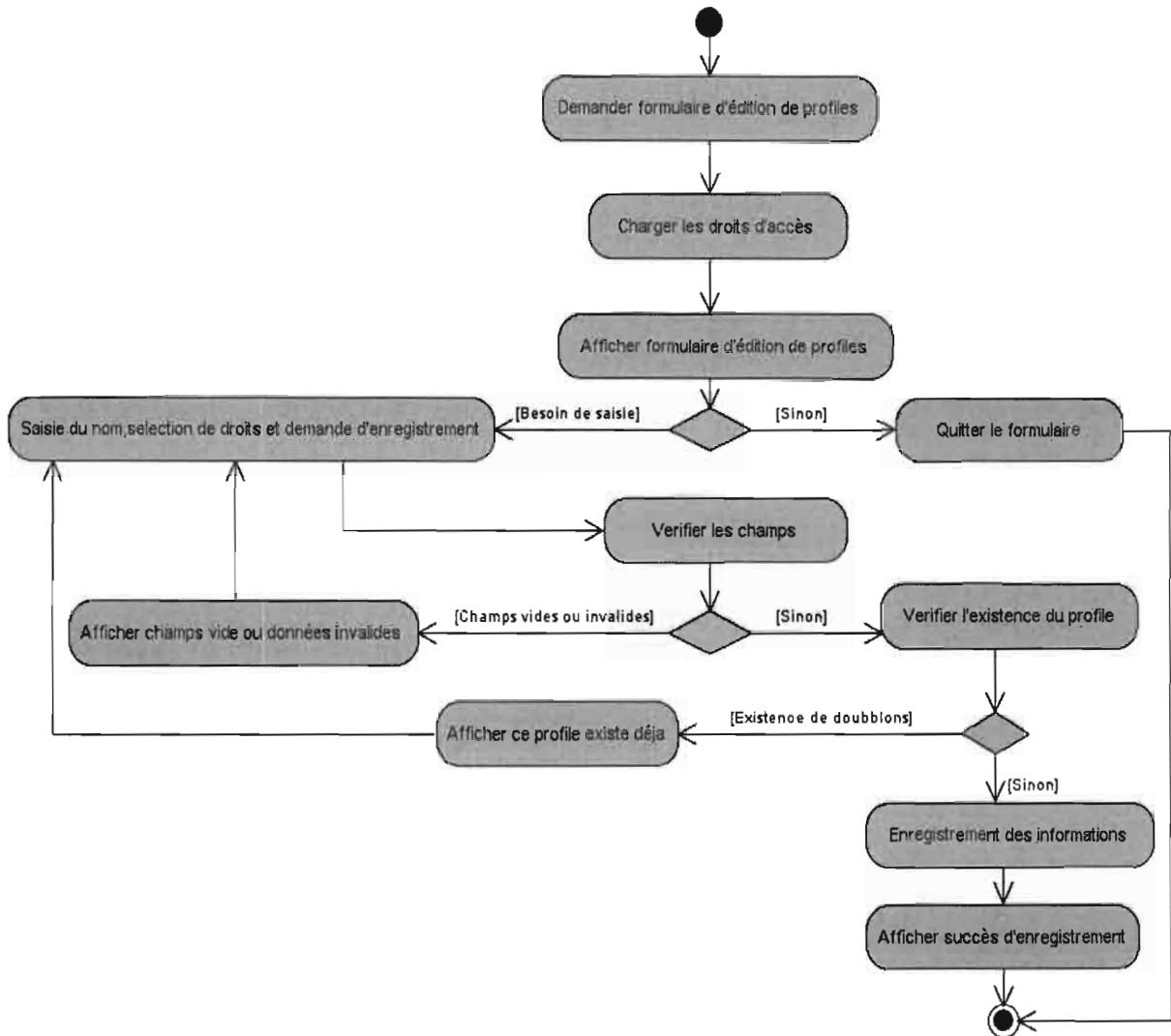
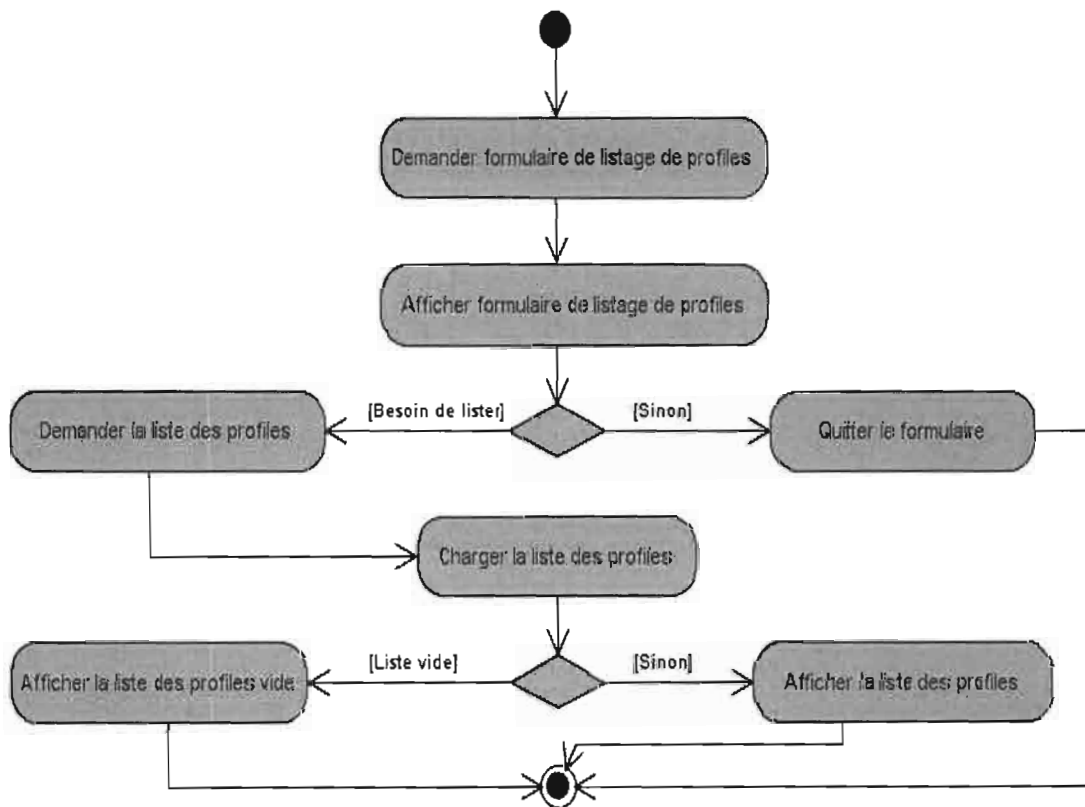


Figure 20 : Diagramme d'activités du cas d'utilisation « Ajouter profile d'utilisateur »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **DIAGRAMMES D'ACTIVITES DU CAS D'UTILISATION « Lister profile d'utilisateur »**

Visual Paradigm for UML Community Edition [not for commercial use]



**Figure 21 : Diagramme d'activités du cas d'utilisation « Lister profile d'utilisateur »**

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **DIAGRAMMES D'ACTIVITES DU CAS D'UTILISATION « Modifier profile d'utilisateur »**

Visual Paradigm for UML Community Edition (not for commercial use)

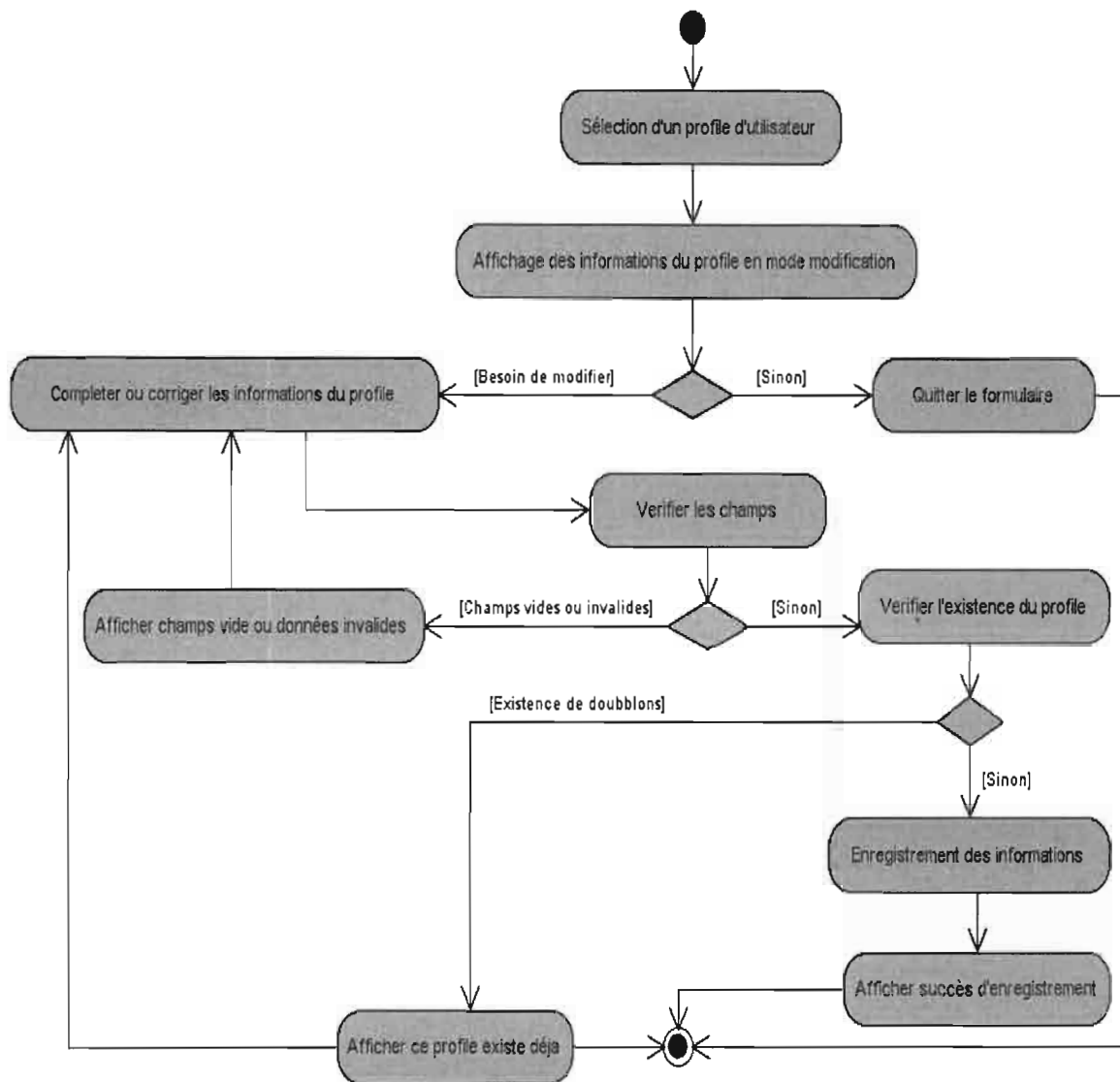


Figure 22 : Diagramme d'activités du cas d'utilisation « Modifier profile d'utilisateur »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **DIAGRAMMES D'ACTIVITES DU CAS D'UTILISATION « Supprimer profile d'utilisateur »**

Visual Paradigm for UML Community Edition [not for commercial use]

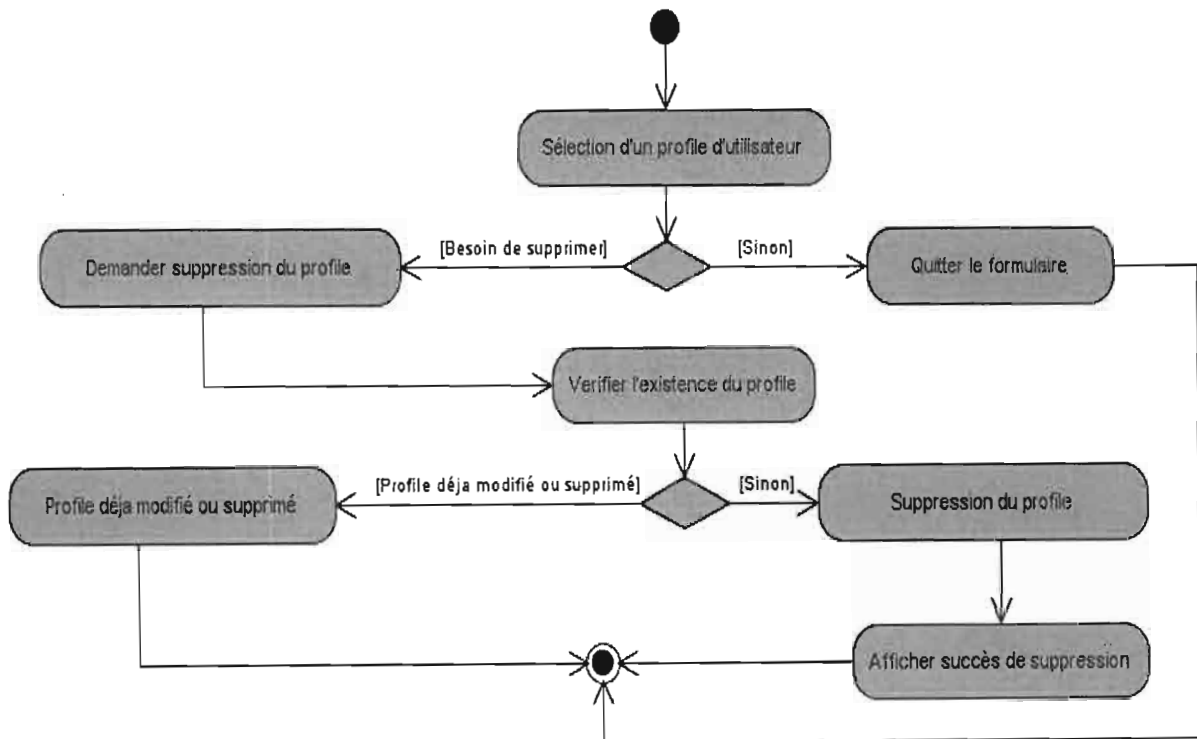


Figure 23 : Diagramme d'activités du cas d'utilisation « Supprimer profile d'utilisateur »

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **DIAGRAMMES D'ACTIVITES DU CAS D'UTILISATION « Viser engagement »**

Visual Paradigm for UML Community Edition (not for commercial use)

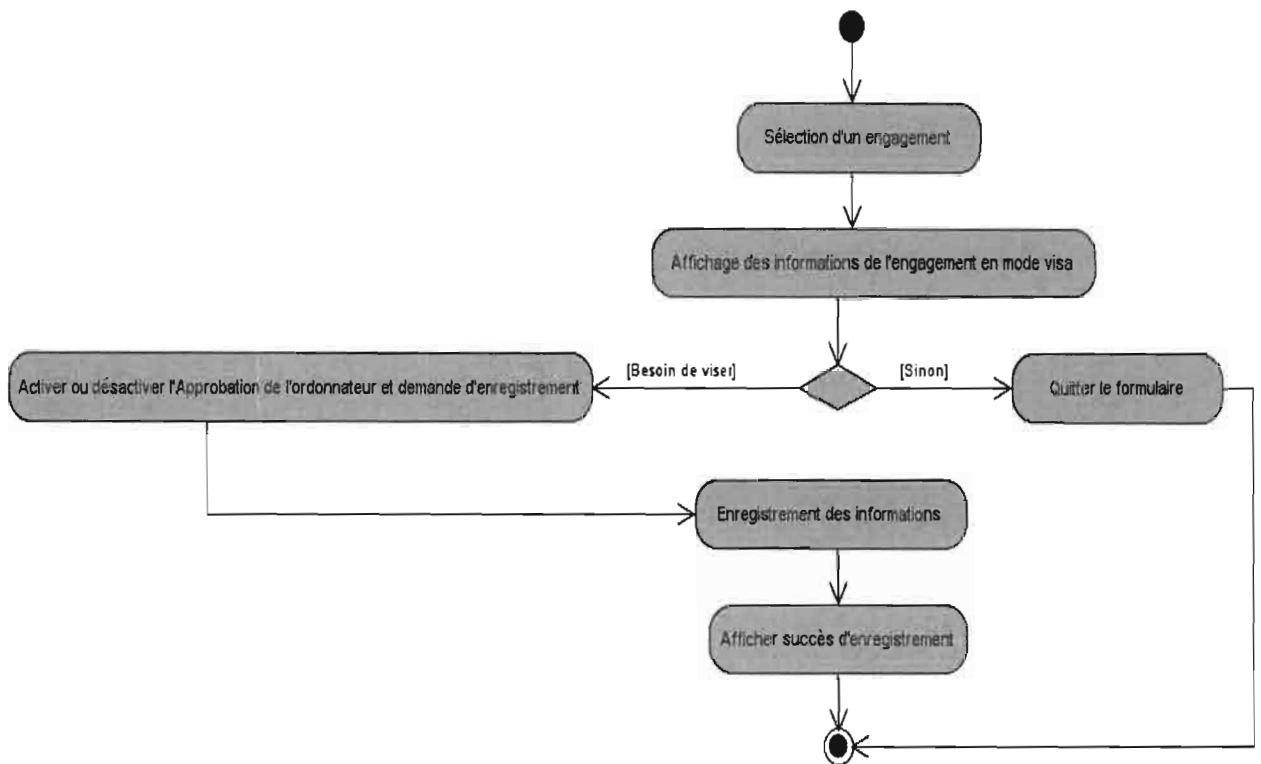


Figure 24 : Diagramme d'activités du cas d'utilisation « Approbation ordonnanceur »

## 1.4 MODELE DE DEPLOIEMENT

L'objectif du modèle de déploiement est de spécifier la configuration physique des matériels (poste de travail, serveurs, etc.), de représenter la façon dont les différents éléments du système doivent être déployés et de décrire les artefacts supportés par chaque matériel.

Nous présentons notre modèle de déploiement dans le schéma ci-dessous :

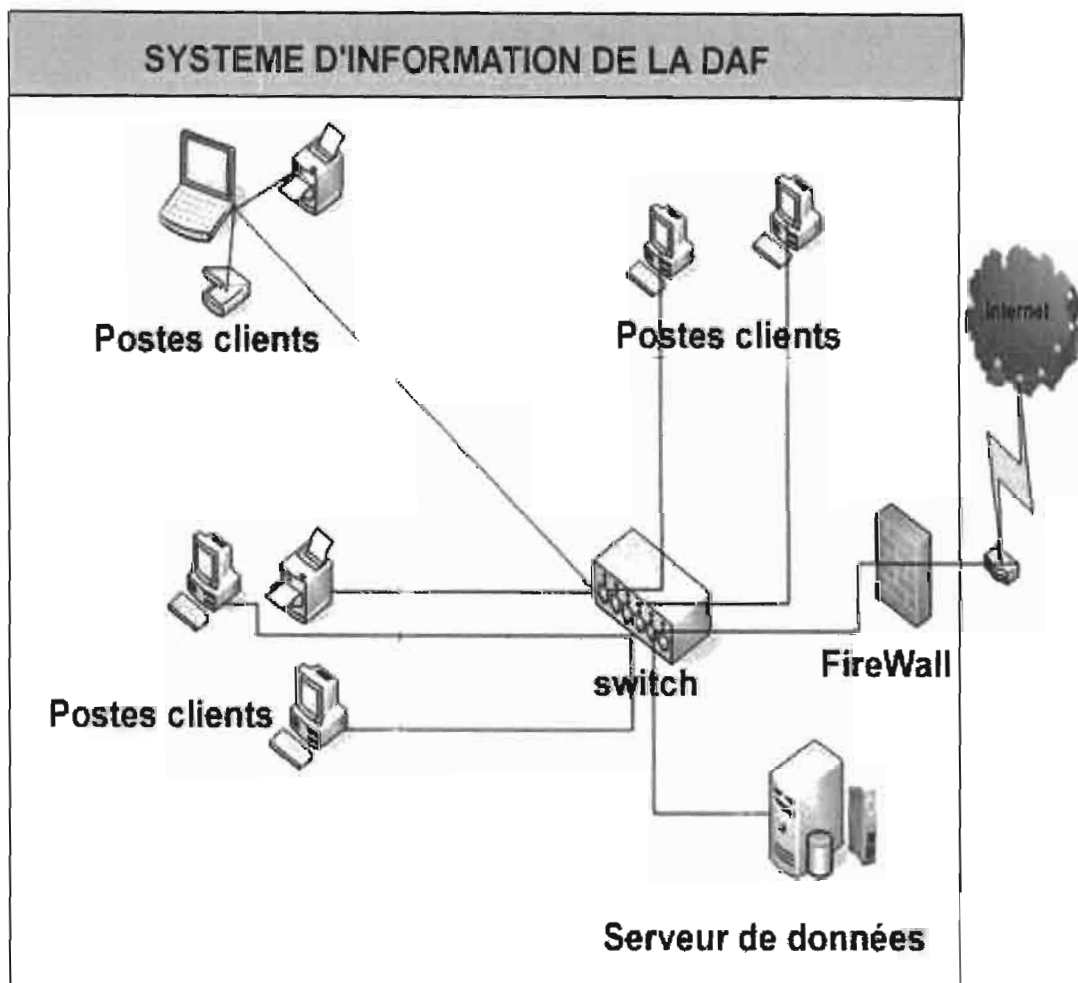


Figure 25 : Schéma du modèle de déploiement

## 2. POLITIQUE DE TRANSITOIRE

La procédure transitoire est un ensemble de tâches consécutives à exécuter pour passer du système actuel au futur système. Le nouveau système sera soumis à une série de tests (correctifs, adaptifs) afin de déceler d'éventuelles erreurs et de s'assurer de son efficacité en terme des services rendus. Les défaillances constatées feront l'objet de correction. Ainsi le nouveau système fonctionnera en parallèle avec l'existant pendant une période déterminée avant que le nouveau système ne prenne entièrement la main.

Ci-dessous nous décrivons les procédures de test adoptées :

- **TESTS UNITAIRES**

En programmation informatique, le test unitaire est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un



# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

programme appelée « unité » ou « module ». Il s'agit pour le programmeur de tester un module, indépendamment du reste du programme, ceci afin de s'assurer qu'il répond aux spécifications fonctionnelles et qu'il fonctionne correctement en toutes circonstances. Un environnement préparé (framework) pour réaliser facilement des tests unitaires existe dans les principaux langages de programmation, on par exemple pour NUnit pour .NET , NUnitASP pour ASP.NET (support interrompu depuis le 31 janvier 2008) .

## • TESTS D'INTEGRATION

Un **test d'intégration** est un test qui se déroule dans une phase d'un projet informatique suivant les tests unitaires. Il consiste, une fois que les développeurs ont chacun validé leurs développements ou leurs correctifs, à regrouper leurs modifications ensemble dans le cadre d'une livraison. Il s'agit d'établir une nouvelle version, basée soit sur une version de maintenance, soit sur une version de développement. L'intégration fait appel en général à un système de gestion de versions, et éventuellement à des programmes d'installation.

L'intégration continue est la fusion des tests unitaires et des tests d'intégration, car le programmeur détient toute l'application sur son poste et peut donc faire de l'intégration tout au long de son développement.

## • TESTS DE VALIDATION

Le test de validation permet de vérifier si toutes les exigences client décrites dans le document de spécification d'un logiciel, écrit à partir de la spécification des besoins, sont respectées. Les tests de validation se décomposent généralement en plusieurs phases:

- **Validation fonctionnelle** Les tests fonctionnels vérifient que les différents modules ou composants implémentent correctement les exigences client. Ces tests peuvent être de type valide, invalide, inopportuns, etc.
- **Validation solution** Les tests solutions vérifient les exigences client d'un point de vue cas d'utilisation. Généralement ces tests sont des tests en volume. Chaque grand cas d'utilisation est validé isolément; puis tous les cas d'utilisation sont validés ensemble.
- **Validation performance, robustesse** Les tests de performance vont vérifier la conformité de la solution par rapport à ses exigences de performance, alors que les tests de robustesse vont essayer de mettre en évidence des éventuels problèmes de stabilité et de fiabilité dans le temps.

## • TESTS DE RECETTE

En informatique, la recette ou test d'acceptation est une phase de développement des projets, visant à assurer formellement que le produit est conforme aux spécifications. Cette étape implique, en la présence effective des différents acteurs du projet, maîtrise d'œuvre et maîtrise d'ouvrage, le déroulement rigoureux de procédures de tests préalablement décrits, et l'identification de tout écart fonctionnel ou technique. La procédure de recette se déroule en deux étapes principales :

- La **recette usine** : comprend tous les tests réalisés chez le fournisseur, avant la livraison. Elle désigne donc les tests unitaires, les tests de validation et les tests d'intégration.
- **La recette utilisateur, VA ou VABF**

Lors de l'étape de vérification d'aptitude (VA) ou vérification d'aptitude au bon fonctionnement (VABF) ou recette utilisateur, le client réalise deux catégories de tests différentes :

- ✓ **La recette fonctionnelle** a pour but la validation des fonctionnalités exprimées dans le cahier des charges et détaillées dans les spécifications fonctionnelles.
- ✓ **La recette technique** Chargée de contrôler les caractéristiques techniques du produit livré, la recette technique regroupe les tests suivants :
  - les tests d'exploitabilité et en particulier le respect des exigences d'architecture technique ;
  - les tests de performance.

## 3. POLITIQUE DE SECURITE

La politique de sécurité a pour but de minimiser les risques de panne, d'éviter que la base de données soit dans un état d'incohérence, d'éviter les accès non autorisés à la base et d'éviter la présence de programmes indésirables dans le réseau.

### 3.1 PROTECTION CONTRE LES CATASTROPHES

Pour la protection contre les catastrophes nous suggérons que : Que la salle soit protégée contre la pluie, le feu et tout autre élément pouvant être source de catastrophe naturelle.

## **3.2 PROTECTION CONTRE LES VIRUS INFORMATIQUES**

Afin de protéger la protection contre les virus, nous suggérons :

- L'utilisation de KAPERSKY comme antivirus sur tous les machines afin d'assurer la protection contre les virus
- D'éviter l'utilisation de clef ou disque dur externe sur le serveur autre que ceux dédiés aux sauvegardes et restaurations de données sur le serveur
- D'éviter également l'utilisation des supports de sauvegardes et restauration sur d'autres machines

## **3.3 PROTECTION CONTRE LES PANNES D'ELECTRICITE**

Nous suggérons l'utilisation d'onduleur pour la protection contre les pannes d'électricité pour tous les postes de travail sans exception du système d'information.

## **3.4 CONFIDENTIALITE DES DONNEES**

Afin d'assurer la confidentialité :

- tout utilisateur doit s'authentifier avant d'accéder à son espace de travail,
- les droit d'accès sont donnés en fonction des profiles accordé à chaque utilisateur,
- les mots de passe sont cryptés
- et nous envisageons de crypter les données avant leur transit sur le réseau

## **3.5 PROTECTION DES DONNEES**

Pour la protection de données :

- La machine qui servira de serveur de données, sur laquelle notre base de données oracle sera hébergée soit dans un local ou l'accès n'est autorisé qu'aux administrateurs du système d'information

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

- Que le local soit physiquement sécurisé, qu'aucune personne dont l'accès à la salle n'est pas autorisé ne puisse y pénétrer
- Nous envisageons une sauvegarde régulière par semaine par exemple afin de minimiser les pertes de données en cas de panne majeure moyennant une sauvegarde récente
- Nous envisageons l'utilisation d'un anti virus firewall afin d'empêcher toute intrusion en filtrant l'accès au réseau d'information.

## **4. PROCEDURES DE SECOURS**

La procédure de secours permet de trouver des solutions palliatives en cas de situation de panne ou de problèmes imprévus.

### **4.1 PANNE DE SERVEUR**

Nous suggérons une impression des états de manière journalière afin de garder un bilan journalier sur papier. Ce qui permet également de savoir les dernières références des dossiers et de poursuivre le travail manuellement en remplissant des fiches vierges en attendant le dépannage du serveur.

### **4.2 POSTE DE TRAVAIL INDISPONIBLE**

Nous suggérons de poursuivre le travail manuellement en remplissant des fiches vierges en attendant la disponibilité du poste de travail.

## CHAPITRE 3 : REALISATION

Dans ce chapitre, nous présentons les outils de modélisation et de développement de notre application, puis nous feront un bilan du stage.

### **1. OUTILS DE MODELISATION**

- **VISUAL PARADIGM FOR UML 6.3 ENTERPRISE EDITION**

Visual Paradigm est un logiciel de modélisation permettant la modélisation des traitements informatiques et des bases données. Nous avons utilisé cet outil pour modéliser nos différents diagrammes présentés dans le rapport.

- **ORACLE XE**

Oracle XE, Système de Gestion de Base de Données (SGBD), utilisé pour stocker nos données. Oracle XE comporte des outils permettant de concevoir des schémas de base de données, d'exécuter des requêtes, d'administrer les bases et leur utilisateurs, programmer des fonctions et procédures stockées.

### **2. LES OUTILS DE DEVELOPPEMENT**

- **VISUAL STUDIO 2010 UTIMATE**

Visual Studio est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). Ce environnement nous a permis de créer nos interfaces rapidement avec un minimum de codage manuel. Il a également permis de programmer aisément nos différents modules et aussi de faire la jonction entre ces modules de notre application. Il fournit également des outils pour tester, déboguer et déployer des

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

applications, incluant une importante bibliothèque de composants réutilisables, un ensemble d'outils de conception, ainsi que des experts de programmation. Ces outils de prototypage et réduisent la durée du développement.

- **ORACLE DATA PROVIDER FOR.NET**

Oracle Data Provider for.Net est un pilote permettant une communication directe entre les applications .net et le SGBD Oracle sans passer par ODBC. Il a l'avantage d'accélérer les requêtes minimisant ainsi les temps de réponses lors de l'envoi des requêtes par l'application.

- **VB.NET**

VB.NET est un langage objet du dot.net que nous avons utilisé pour programmer notre application, vue sa puissance et sa souplesse.

- **TIAK IDE**

TIAK IDE est une application personnelle que nous avons développée qui est la somme de l'expérience que nous avons obtenue à travers le développement d'applications de gestion que nous avons réalisés avec VB.NET et adapté à notre style de programmation. Il permet de générer du code source permettant d'insérer, lister, supprimer, rechercher des données dans la base de données, donc permet de générer le code source nécessaire pour l'accès aux données. Il génère également du code source permettant la jonction entre l'interface conçu sous Visual Studio et code source d'accès données. Les codes sources générés peuvent directement être copié et utilisé sous Visual studio avec possibilité de personnaliser encore mieux.

En somme TIAK IDE nous a permis d'accélérer la réalisation de notre application, en nous dispensant d'une bonne partie de programmation manuelle.

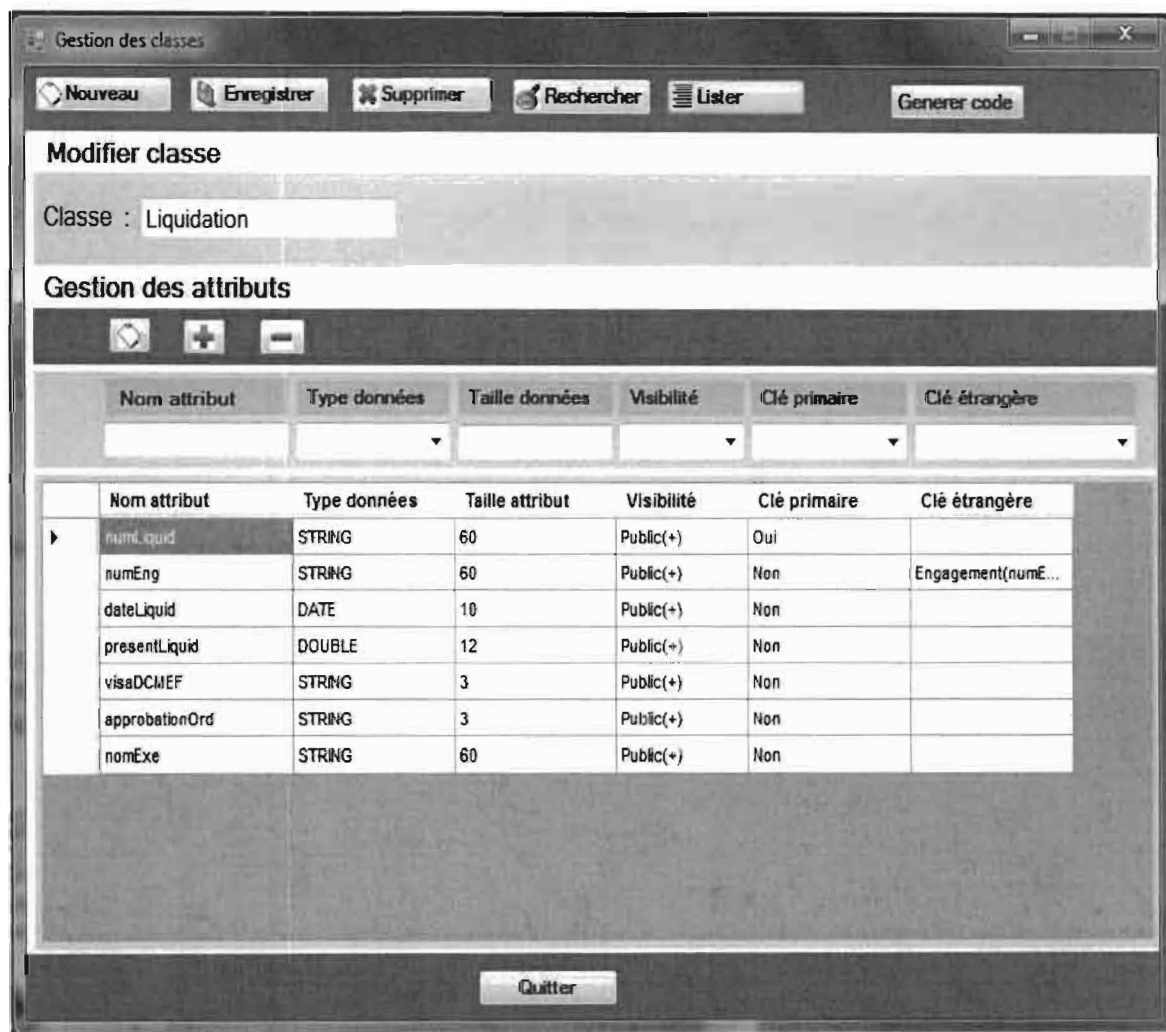


Figure 26 : Aperçu de l'interface de TIAK IDE

### 3. QUELQUES CAPTURES D'ECRAN

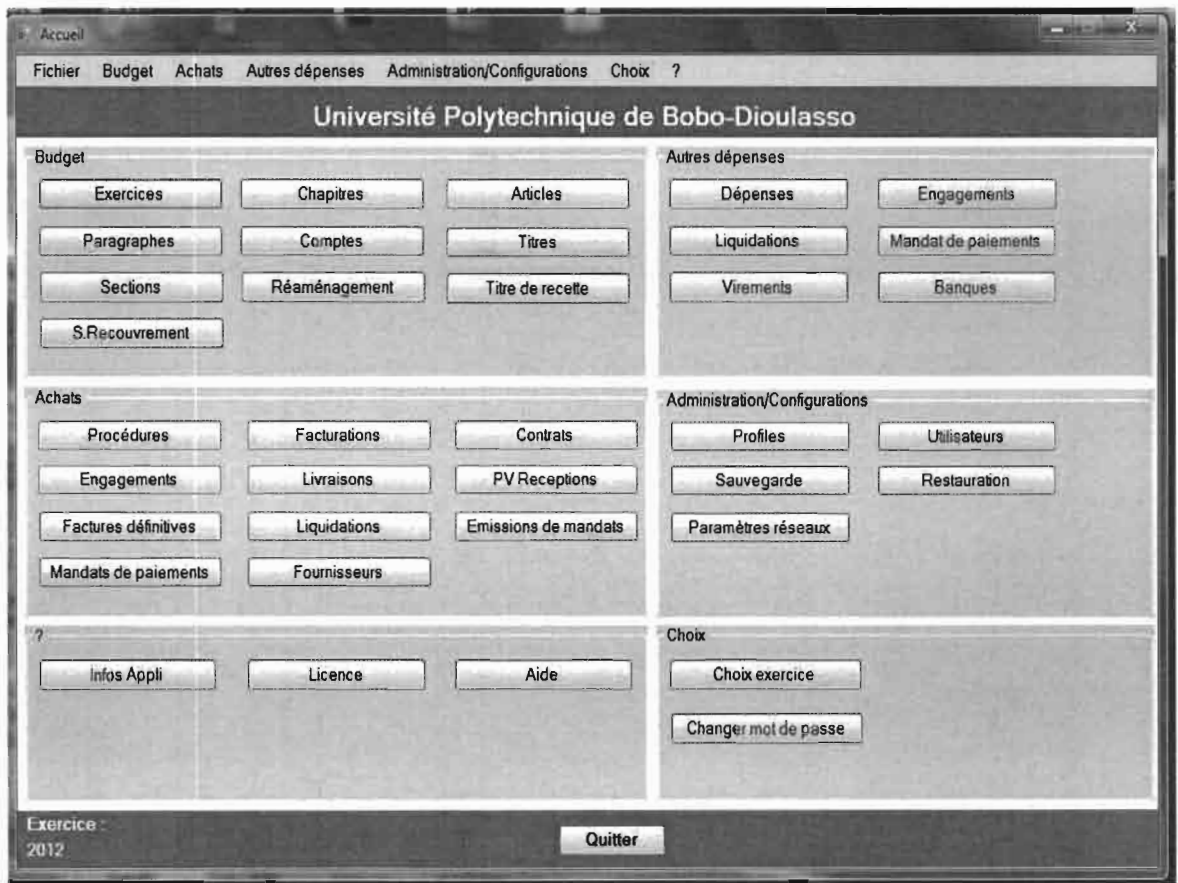


Figure 27 : Aperçu de la page d'accueil

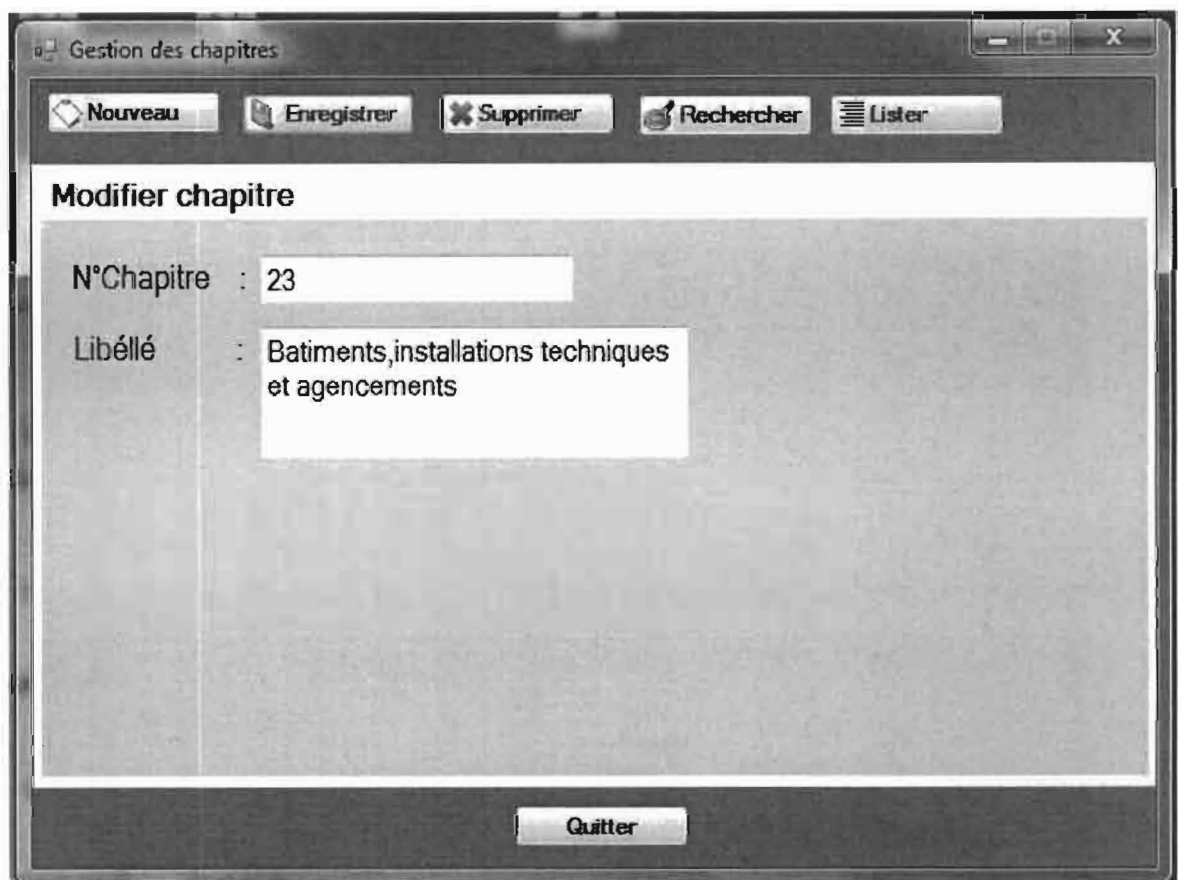


Figure 28 : Aperçu de la page d'édition de chapitres



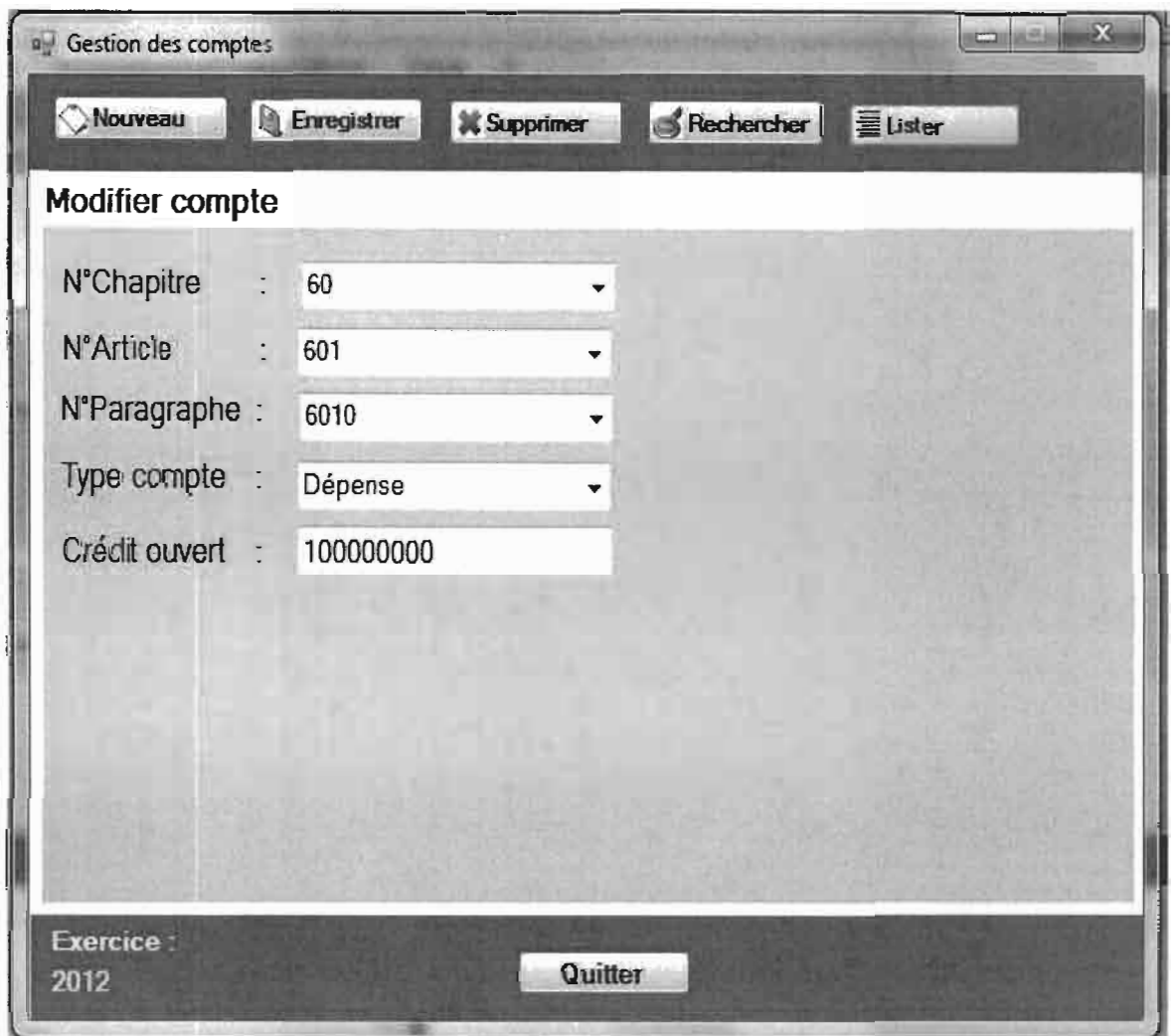


Figure 29 : Aperçu de la page d'édition de comptes

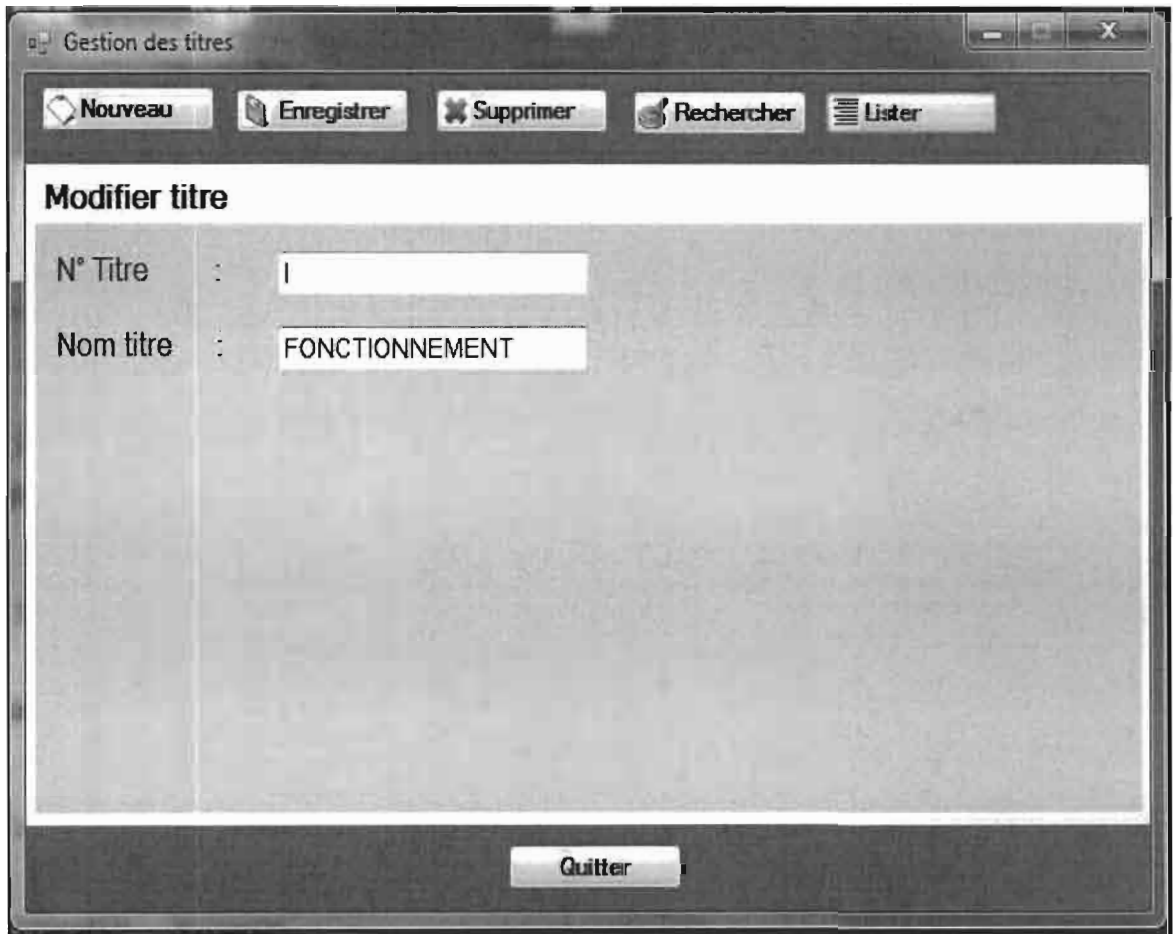


Figure 30 : Aperçu de la page d'édition de titres

The screenshot shows a software application window titled "Gestion des demandes de prix". At the top, there is a menu bar with five buttons: "Nouveau", "Enregistrer", "Supprimer", "Rechercher", and "Lister". Below the menu bar, the main content area is titled "Modifier demande de prix". This area contains a form with the following fields:

- N°Demande de prix : 1
- Type de demande : Demande de quotation (dropdown menu)
- Date début : 25/04/2014 (calendar icon)
- Description : DIVERS

At the bottom of the window, there is a footer area with the text "Exercice : 2012" on the left and a "Quitter" button on the right.

Figure 31 : Aperçu de la page d'édition de demande de prix

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

Gestion des propositions de prix

Nouveau Enregistrer Supprimer Rechercher Lister Imprimer

**Nouvelle proposition de prix**

N°Ordre : 2 N°FU fournisseur :  Inserir

Reffacture :  Sélection :  Oui  Non

Date émission : 30/04/2014  D

N°Demande de prix :  Inserir

**Liste des articles proposés**

Nouveau Ajouter Retirer

Produits	Quantité	Prix U	TOTAL

Désignation Quantité Prix U Total

Total général :

Complements d'informations

Exercice 2012 Quitter

Figure 32 : Aperçu de la page d'édition de proposition de prix

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

The screenshot displays a web application window titled "Gestion des contrats". At the top, there is a navigation bar with buttons for "Nouveau", "Enregistrer", "Supprimer", "Rechercher", "Lister", and "Imprimer".

The main content area is divided into two sections:

- Nouveau contrat:** This section contains a form with the following fields:
  - N°Contrat: Text input field.
  - Type de contrat: Dropdown menu.
  - N°Ordre facture proforma: Text input field with an "Insérer" button.
  - Réf de facture proforma: Text input field.
  - Approuvé le: Date field (06/07/2014) with a "D" button.
  - Notifié le: Date field (06/07/2014) with a "D" button.
  - Délais d'exécution: Text input field followed by "(Jours)".
  - Visa DCMEF: Radio buttons for "Oui" and "Non" (checked).
  - Approbation Ordonnateur: Radio buttons for "Oui" and "Non" (checked).
- Liste des articles commandés:** This section features a table with columns "Produits", "Quantité", "Prix U", and "TOTAL". Above the table are buttons for "Nouveau", "+ Ajouter", "- Retirer", and "Charger". Below the table, there is a "Total général:" label and an input field. The table content is currently empty.

At the bottom left, the text "Exercice 2012" is visible. At the bottom right, there is a "Quitter" button.

Figure 33 : Aperçu de la page d'édition de contrats

The screenshot shows a software window titled "Gestion des engagements". At the top, there is a menu bar with buttons for "Nouveau", "Enregistrer", "Supprimer", "Rechercher", and "Lister". Below this is a section titled "Modifier engagement".

The form contains the following fields and controls:

- N'Engagement : 1
- N'Compte d'imputation : 6010 [Inserer]
- N'Contrat : 1 [Inserer]
- Objet : Bon de commande
- Titre : 1
- Section : 1
- Date engagement : 12/05/2014 [D]
- Present engagé : 525000 [Charger]
- Visa DCMEF :  Oui  Non
- Signature DAF :  Oui  Non
- Validation Ordonnateur :  Oui  Non

Below the main form is a section titled "Complements d'informations" with the following fields:

- Crédits ouverts : 100000000
- Crédits engagés : 0
- Total engagement : 525000
- Crédit disponible : 99475000

At the bottom left, it says "Exercice : 2012" and at the bottom right, there is a "Quitter" button.

Figure 34 : Aperçu de la page d'engagements

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

**Gestion des liquidations**

Nouveau Enregistrer Supprimer Rechercher Lister

### Modifier liquidation

N°Liquidation : 3

N°Facture définitive : 1 **Inserer**

Reffacture : 1 DU 12-05-2012

N°Engagement : 1

Compte imputation : 6010

Date liquidation : 06/07/2014 **D**

Present liquidé : 225000 **Charger**

Visa DCMEF :  Oui  Non

Validation ordonnateur :  Oui  Non

### Complements d'informations

Somme mandatée : 225000

Reste à liquider du présent engagement : 300000

Total général liquidé sur le compte : 725000

Disponible sur le compte : 99275000

Exercice : 2012 **Quitter**

Figure 35 : Aperçu de la page d'édition de liquidations

The screenshot shows a software application window titled "Gestion des émissions de mandats". At the top, there is a menu bar with five buttons: "Nouveau", "Enregistrer", "Supprimer", "Rechercher", and "Lister". Below the menu bar, the main content area is titled "Modifier bordereau d'émission de mandat". This area contains several form fields and buttons:

- N'Bordereau d'émission : 1
- N'Facture définitive : 1
- Refacture : 1 DU 10-03
- N'Engagement : 1
- Compte imputation : 6015
- Date mandat : 29/04/2012
- Montant mandaté : 1000000
- Visa DCMEF :  Oui  Non
- Visa Ordonnateur :  Oui  Non

At the bottom of the window, there is a footer area with the text "Exercice : 2012" and a "Quitter" button.

Figure 36 : Aperçu de la page d'émission de mandats



The screenshot shows a window titled "Mandat de paiement" with a toolbar containing buttons for "Nouveau", "Enregistrer", "Supprimer", "Rechercher", and "Lister". The main area is titled "Modifier mandat de paiement" and contains the following fields:

N°Mandat paiement	:	1	
N°Facture définitive	:	1	<input type="button" value="Insérer"/>
Ref facture	:	1 DU 10-03-2012	
N°Engagement	:	1	
Compte imputation	:	6015	
Date mandat	:	29/04/2012	<input type="button" value="D"/>
Montant mandaté	:	1000000	<input type="button" value="Charger"/>
Type règlement	:	Espèce	<input type="button" value="v"/>
Visa DCMEF	:	<input type="checkbox"/> Oui	<input checked="" type="checkbox"/> Non
Visa Ordonnateur	:	<input type="checkbox"/> Oui	<input checked="" type="checkbox"/> Non
Visa CC :	:	<input type="checkbox"/> Oui	<input checked="" type="checkbox"/> Non

At the bottom, there is a label "Exercice : 2012" and a "Quitter" button.

Figure 37 : Aperçu de la page d'édition de mandats de paiements

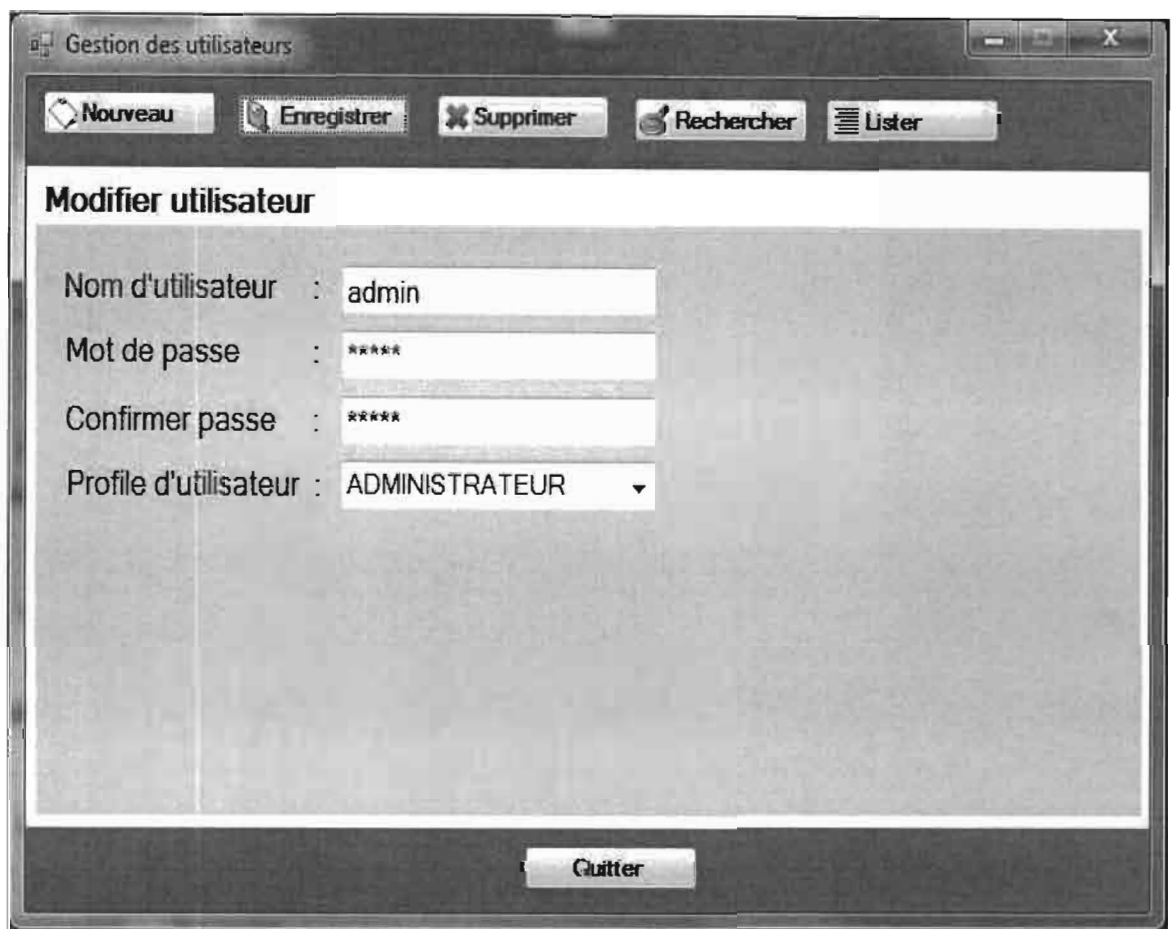


Figure 38 : Aperçu de la page d'édition de comptes d'utilisateurs

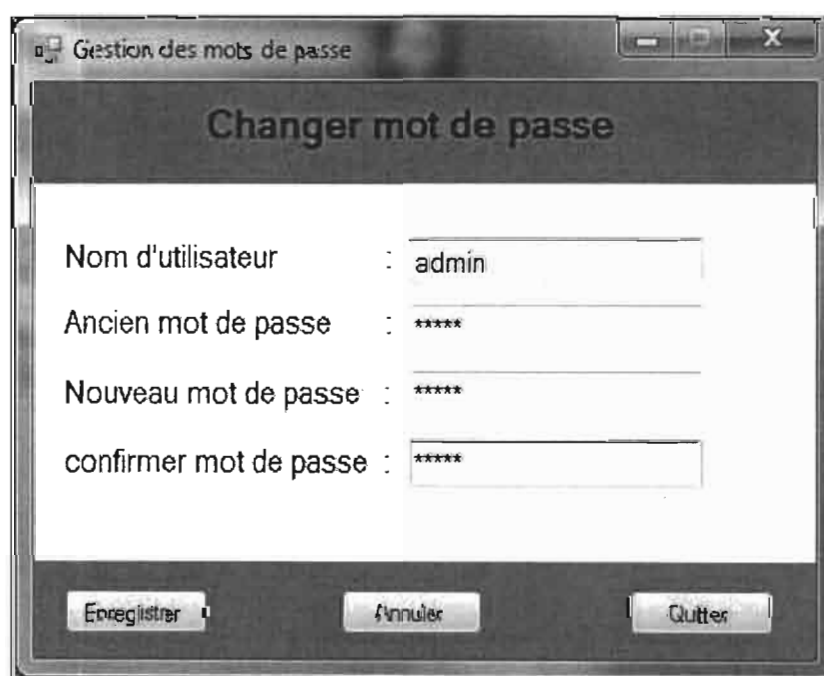


Figure 39 : Aperçu de la page de gestion de mot de passe

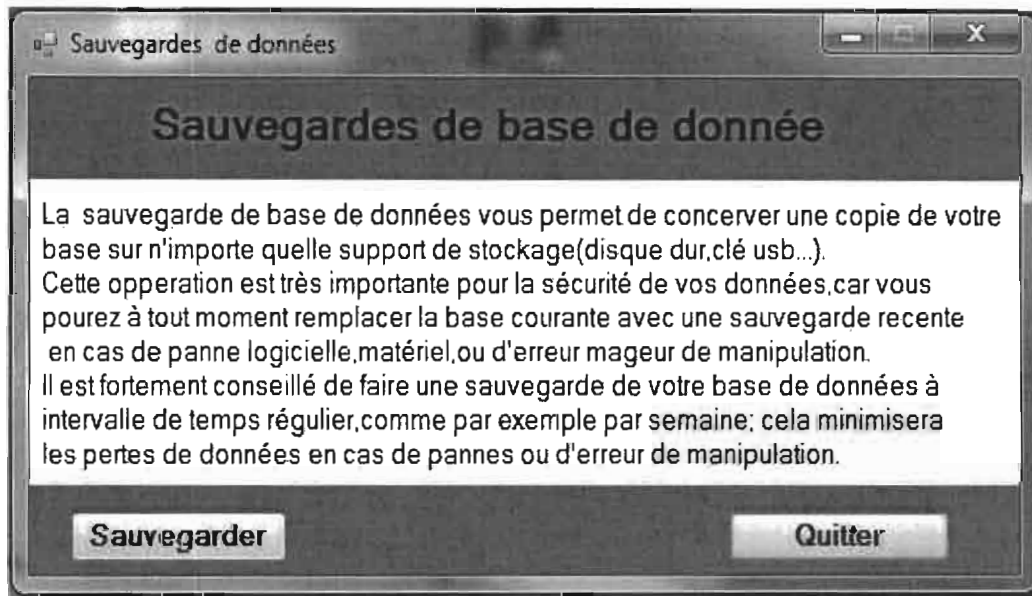


Figure 40 : Aperçu de la page d'édition de sauvegarde de données

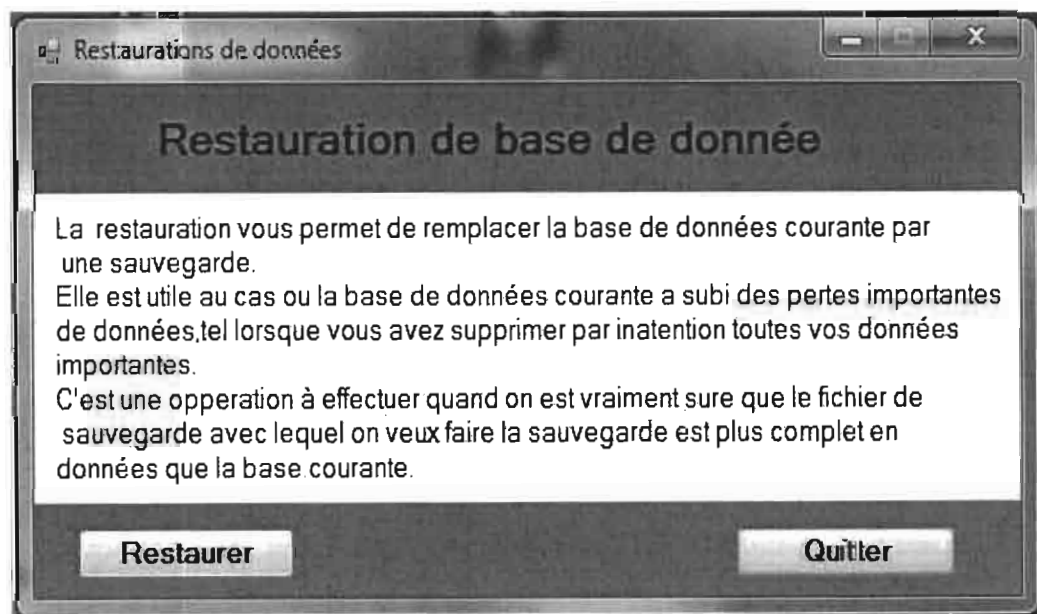


Figure 41 : Aperçu de la page d'édition de restaurations de données

## 4. BILAN DU STAGE

### 4.1 PLANNING REEL

Nous avons essayé d'effectuer le développement selon le planning prévisionnel à la fin des délais du planning prévisionnel le projet était toujours inachevé. Jusqu'au mois de Juillet

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

2014 le projet a suivi son cours suivant le cycle de vie en V utilisé, nous effectuons les évolutions suivant une procédure incrémentale.

## **4.2 ANALYSE DES ECARTS**

Nous constatons qu'il y a un écart net entre le planning prévisionnel et le planning réel. Ceci s'explique par le fait le projet a été exécuté par un groupe de projet plus réduit (une personne) que celui requis (six personnes) et que les délais théoriques dans le planning prévisionnel on été établie en conformité avec les délais du temps de stage ce qui est nettement plus court que celui décrit dans les calcul d'estimation des délais et coût de développement.

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## **CONCLUSION GENERALE**

Notre projet a consisté en l'analyse la conception et la réalisation d'un outil de gestion budgétaire, afin d'apporter une valeur ajoutée et un meilleur service à la DAF de l'UPB.

Dans le premier chapitre nous avons présenté le service, étudié l'existant, étudié les besoins du futur système ; dans le second nous avons regroupé les éléments de domaine d'étude en catégories et procédé à la conception, nous avons présenté les politiques transitoire, de sécurité et de secours ; dans le troisième chapitre nous avons présenté nos outils de modélisation et de développement, présenté quelques captures d'écran et fait un bilan du stage.

Aux termes de notre stage nous estimons la réalisation du projet à 70%, ce projet d'étude nous a permis d'améliorer nos capacités d'analyse et de conception et de mieux comprendre la gestion budgétaire, ce qui nous permet de réaliser un outils de gestion budgétaire au modèle de gestion budget de services publics au Burkina en générale et particulièrement adapté à la gestion de l'UPB.

## **BIBLIOGRAPHIE ET WEBOGRAPHIE**

- **Livre UML2 en action**
- **Mémoire de fin d'étude de Mr Mohamed Anouar DAHDEH de Master Professionnel en Nouvelles Technologies de Télécommunication et de Réseaux**
- **Mémoire de fin d'étude de Cycle d'Ingénieur de Conception en Informatique de Mr Soumaïla Saga**
- **Notre rapport de deuxième année du Cycle des Ingénieurs de Conception en Informatique**
- **[www.labri.fr/~marlet](http://www.labri.fr/~marlet)**
- **[www.developpez.com](http://www.developpez.com)**
- **[www.uml.org](http://www.uml.org)**

## UTILISATION BIBLIOGRAPHIE ET WEBOGRAPHIE

Nous avons utilisé notre bibliographie et webographie comme suit :

Pour l'élaboration de la méthode du cycle vie en V nous nous sommes inspirés du Mémoire de fin d'étude de Mr Mohamed Anouar DAHDEH de Master Professionnel en Nouvelles Technologies de Télécommunication et de Réseaux et du Mémoire de fin d'étude de Cycle d'Ingénieur de Conception en Informatique de Mr Soumaïla Saga.

Pour les détails des activités à mener dans chaque partie du Cycle nous avons beaucoup utilisé Livre UML2 en action, Notre rapport de deuxième année du Cycle des Ingénieurs de Conception en Informatique.

Quand à la webographie elle a été utilisée à tous les niveaux selon le besoins afin de renforcer ou d'éclaircir certain aspects de la modélisation ou du développement.

## ANNEXE

### UML: UNIFIED MODELING LANGUAGE

#### HISTORIQUE

Au début des années 90, une cinquantaine de méthodes d'analyse et de conception objet existaient. Parmi elles, trois étaient considérées comme les plus importantes : BOOCH de Grady Booch, OMT de James Rumbaugh et enfin OOSE d'Ivar Jacobson. Ces trois auteurs ont ensuite décidé d'unir leurs efforts au sein de la société Rational Software et en 1996 la version **0.9** d'Unified Modeling Language (UML) est proposée. Deux éléments importants sont à noter :

- le terme unified signifie que les auteurs ont essayé de regrouper les éléments importants des concepts objets,

- le terme langage montre qu'il s'agit d'un langage de modélisation et non d'une méthode. Les principaux acteurs du secteur informatique ont ensuite participé à cet effort, et

UML **1.0** a été proposé à l'Object Management Group (OMG). Cet organisme international chargé de définir des standards dans le domaine de l'objet normalise UML 1.1 en 1997. Cette norme a depuis continué d'évoluer et nous en sommes aujourd'hui à la norme 2.3 sortie depuis mai 2010. UML est un langage qui permet de modéliser non seulement des applications informatiques ou des structures de données, mais également les activités d'un domaine: mécanique, biologie, processus métier ...

#### **UML en bref**

Ce langage est tout d'abord graphique et basé autour d'entités et de relations.

Contrairement à d'autres formalismes reposant sur un seul type de diagramme. Un modèle est ici représenté selon différents aspects, chacun à l'aide d'un type de diagramme particulier. Il peut être utilisé aussi bien durant les phases d'analyse, de conception, ou d'implémentation.

#### **Les diagrammes**

Ils sont au nombre de treize, de 3 types :

- **les diagrammes de structure**: diagramme de classe, diagramme composite, diagramme de composants, diagramme de déploiement, diagramme d'objets, diagramme de package



# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

- **les diagrammes de comportement:** diagramme d'activité, diagramme de cas d'utilisation, diagramme d'état-transition
- **les diagrammes dynamiques:** diagramme de séquence, diagramme de communication, diagramme global d'interaction, diagramme de temps.

## 1. DEFINITION DU CAS D'UTILISATION (USE CASE)

Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ses utilisateurs. Une fois identifiés et structurés, ces besoins :

- définissent le contour du système à modéliser (ils précisent le but à atteindre),
- permettent d'identifier les fonctionnalités principales (critiques) du système.

### • Acteur

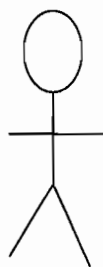
Un acteur est un utilisateur du système. Il peut être :

soit un humain ;

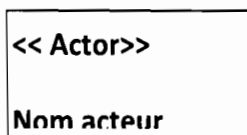
soit un logiciel ;

soit un automate.

On distingue les acteurs physiques et les acteurs non physiques



Un acteur physique



Acteur non physique (Systèmes connexes)

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

---

- **Cas d'utilisation**

Un cas d'utilisation "Représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier" (UML de Pascal Roques, paru en 11/2005).

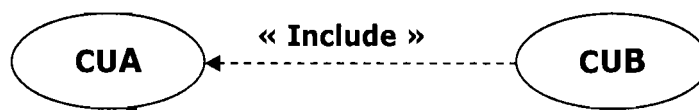
Ils spécifient les fonctionnalités attendues du système. Un cas d'utilisation représente un ensemble de séquences possibles d'actions réalisées par le système et produisant un résultat observable et mesurable afin de satisfaire les objectifs de l'utilisateur du système.



- **Relation entre les cas d'utilisation**

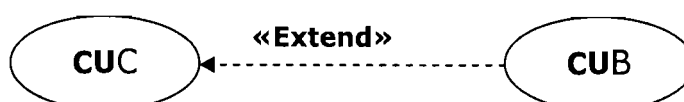
- **Relation d'inclusion (include)**

La relation d'inclusion sert à enrichir un cas d'utilisation par un autre cas d'utilisation. Une relation d'inclusion d'un « cas d'utilisation B » vers un « cas d'utilisation A » indique qu'une instance du « cas d'utilisation B » contient également le comportement spécifié par le « cas d'utilisation A ». Ce comportement est inséré à un endroit défini par le « cas d'utilisation B ». Dans le diagramme des cas d'utilisation, cette relation est représentée par une flèche à traits interrompus muni du stéréotype « include ».

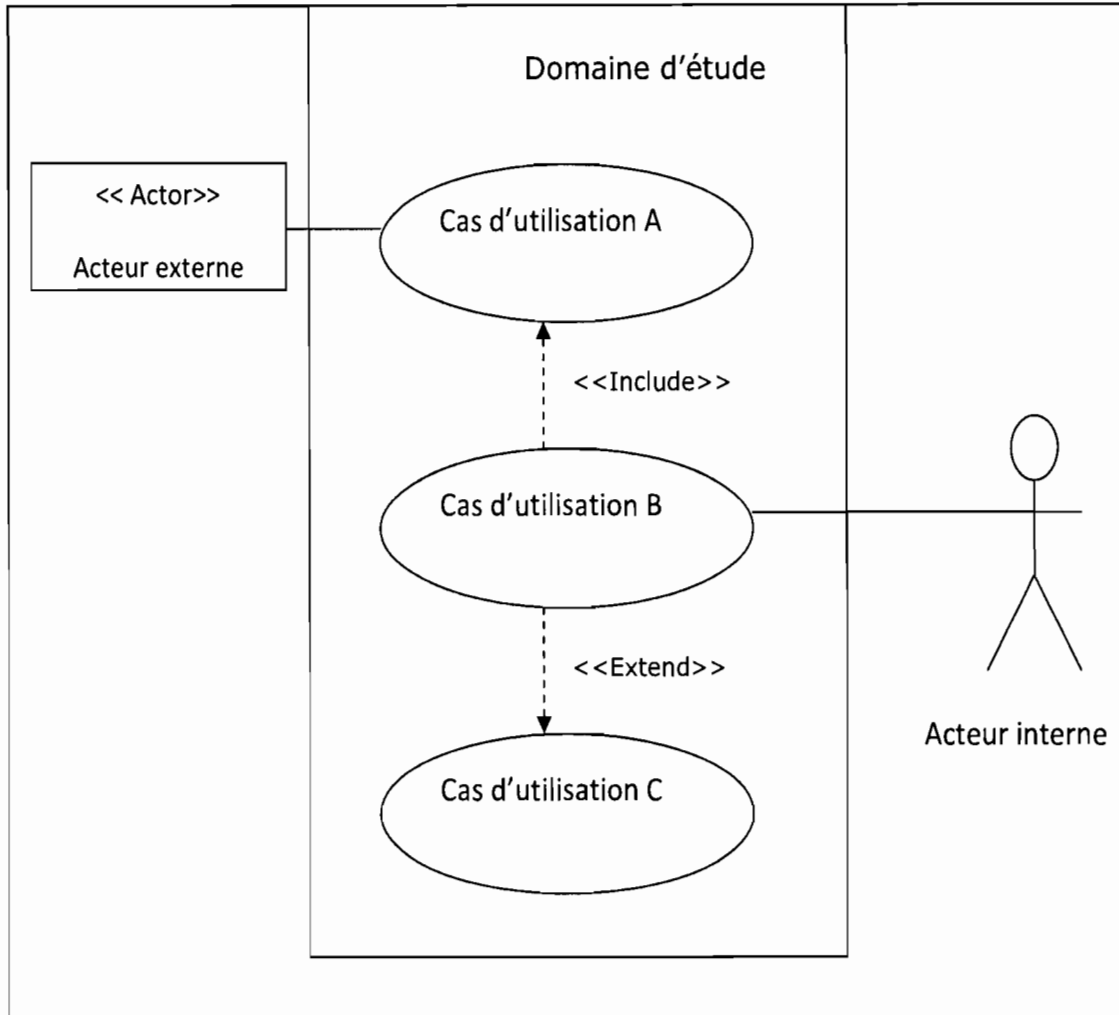


- **Relation d'extension (extend)**

La relation d'extension d'un « cas d'utilisation B » à un « cas d'utilisation C » indique qu'une instance du « cas d'utilisation C » peut être augmentée par le comportement du « cas d'utilisation B ». Le « cas d'utilisation B » est inséré à l'endroit défini par le point d'extension par le « cas d'utilisation C ». Dans le diagramme des cas d'utilisation, cette relation est représentée par une flèche à traits interrompus munis du stéréotype « extend ».



- **Formalisme du diagramme des cas d'utilisation**



**Formalisme : diagramme de cas d'utilisation**

## 2. DIAGRAMMES DE SEQUENCE

Les diagrammes de séquence sont une représentation temporelle des objets et de leurs interactions.

- **Les types de message**

Un message est un moyen de communication entre objets. Ici, le message caractérise un événement c'est-à-dire une information envoyée à un objet et provoquant en réponse le déclenchement d'actions associées à cet objet.

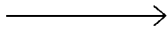
UML propose un certain nombre de stéréotypes graphiques pour décrire la nature du message :

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **message simple**

Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.

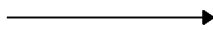
Message simple



- **message minuté (timeout)**

Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.

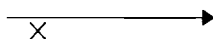
Message minuté



- **message synchrone**

Bloque l'expéditeur jusqu'à la prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise en compte du message.

Message synchrone



- **message asynchrone**

N'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

Message asynchrone



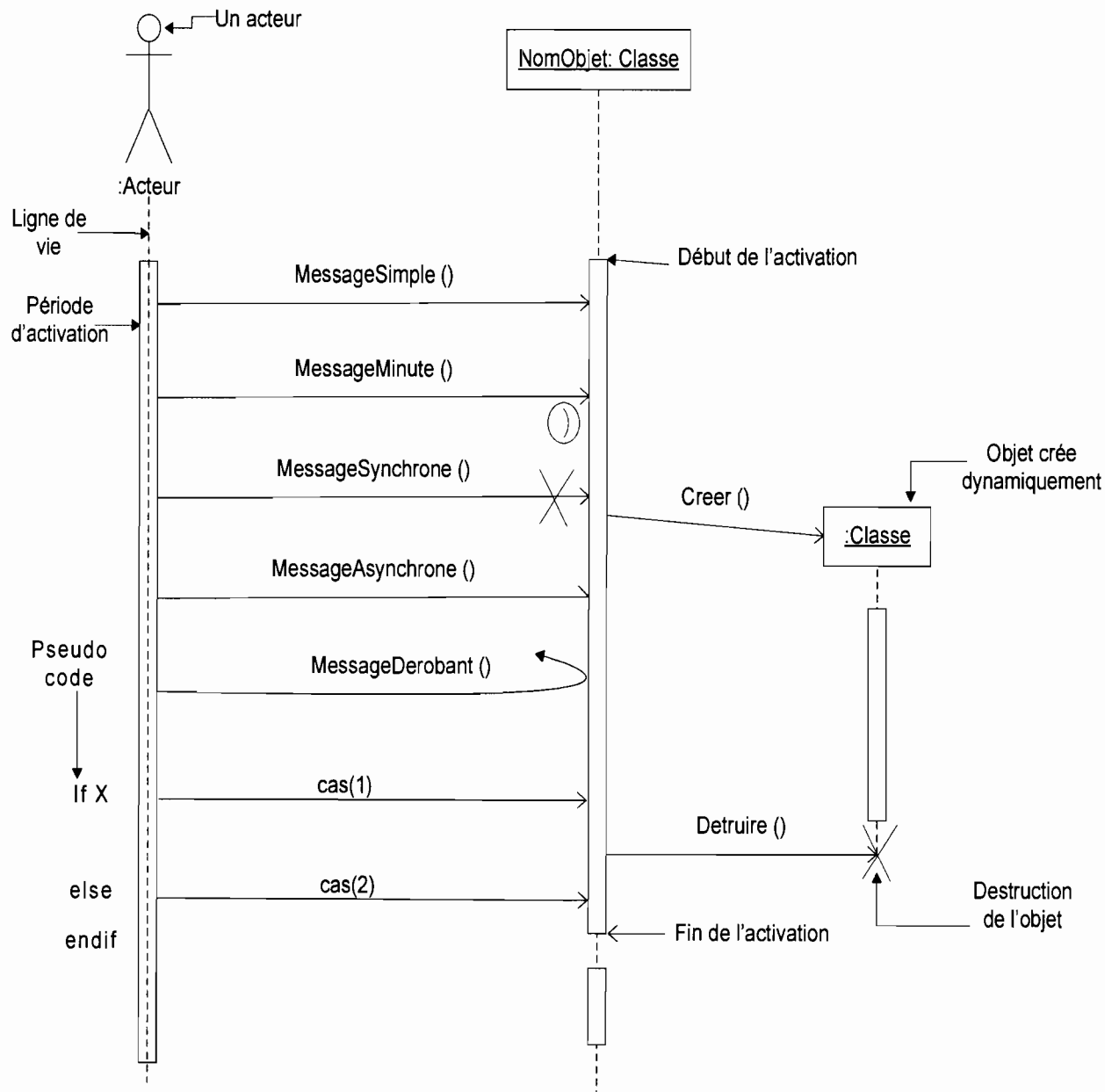
# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **message déroband**

N'interrompt pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message



- **Le formalisme du diagramme de séquence**

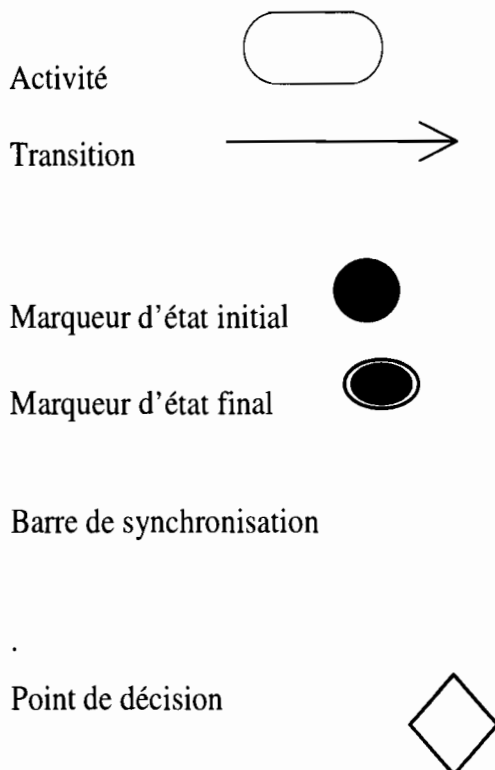


Formalisme : Diagramme de séquence

## 3. DIAGRAMMES D'ACTIVITES

Le diagramme d'activités sert à décrire les flux de contrôle dans un cas d'utilisation ou entre plusieurs cas d'utilisation, décrire le comportement interne d'une classe, décrire le comportement d'une opération

- **Principaux éléments de notation**



- **Transition :**

Une transition matérialise le passage d'une activité vers une autre. Les transitions sont déclenchées par la fin d'une activité et provoquent le début d'une autre (elles sont automatiques).

- **Synchronisation :**

Il est possible de synchroniser les transitions à l'aide des "**barres de synchronisation**" (comme dans les diagrammes d'état transitions). Une barre de synchronisation permet d'ouvrir et de fermer des branches parallèles au sein d'un flot d'exécution :

## Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- les transitions qui partent d'une barre de synchronisation ont lieu en même temps ;
- on ne franchit une barre de synchronisation qu'après réalisation de toutes les transitions qui s'y rattachent.

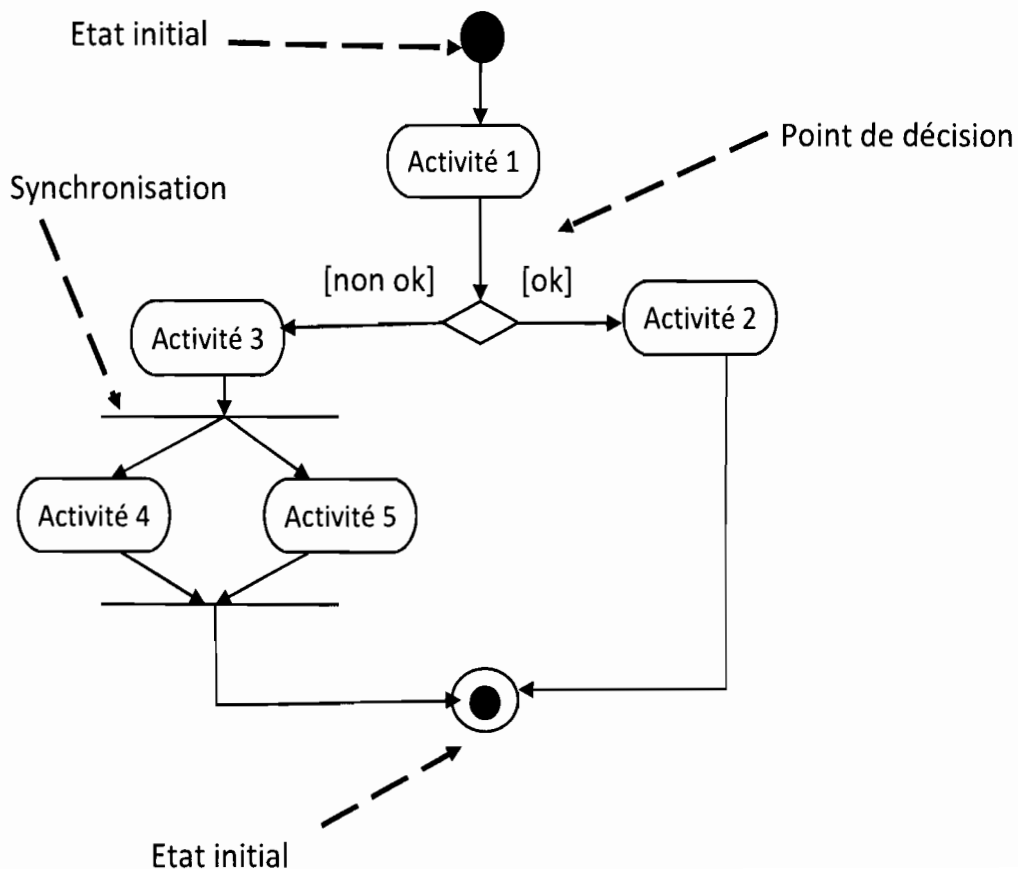
- **Activité :**

Une activité définit un comportement décrit par un séquençement organisé d'unités dont les éléments simples sont les actions

- **Deux pseudo-d'états :**

- État initial: la création de l'instance (indispensable)
- État final: destruction de l'instance (optionnel)

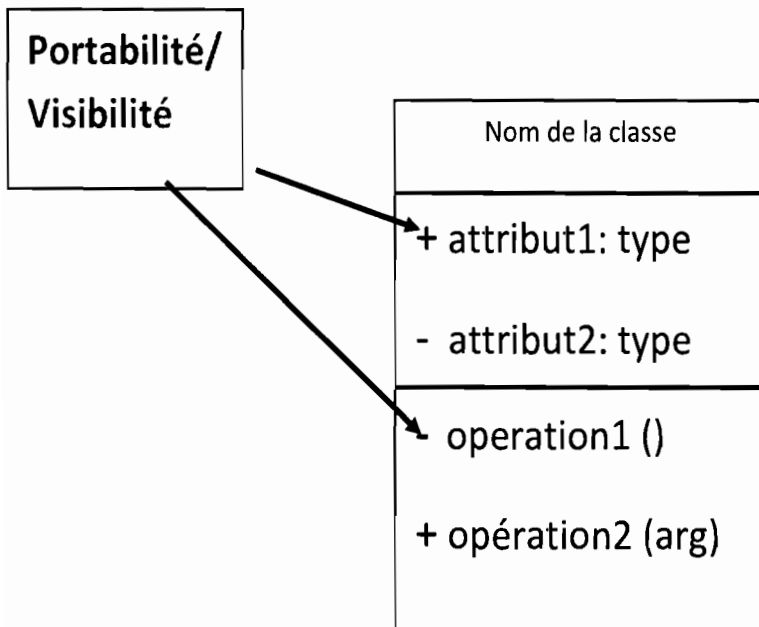
- **Formalisme de description du diagramme d'activité**



## 4. DIAGRAMMES DE CLASSES

Une classe est la description abstraite d'un ensemble d'objets ayant les mêmes caractéristiques (attributs), le même comportement et les mêmes relations sémantiques.

- Représentation d'une classe





# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **Définitions**

**Méthodes** : une méthode ou opération est une fonctionnalité assurée par la classe. Elle spécifie un comportement attendu d'un objet du système. Elle est caractérisée par un nom, un type retourné et éventuellement une liste d'arguments.

**Attributs** : un attribut est une information élémentaire composant une classe. Un attribut peut permettre d'identifier la classe. Il est typé (Integer, Real, String...).

**Visibilités** : UML définit trois niveaux de visibilité pour les attributs et les opérations

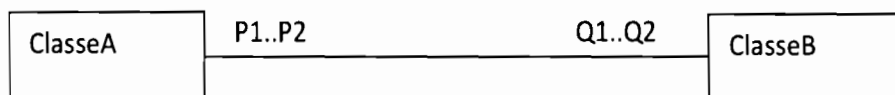
:

Public (+) qui rend l'élément visible à tous les clients de la classe ;

- Protégé (#) qui rend l'élément visible aux sous classes de la classe ;
- Privé (-) qui rend l'élément visible à la classe seule.

**Multiplicité des associations** : chaque rôle peut porter une multiplicité montrant combien d'objets de la classe considérée (celle qui joue ce rôle) peuvent être liés à une instance de l'autre classe par l'association. La multiplicité est représentée sous la forme d'un couple de cardinalités.

1..1 noté 1	Un et un seul
0..1	Zéro ou un
0..* noté *	De Zéro à n
1..*	De un à n
n..m	De n à m



Pour une instance de ClasseA, il y a au minimum **Q1** instance(s) de ClasseB et au maximum **Q2**. De la même façon, pour une instance de ClasseB, il y a au minimum **P1** instances de ClasseA et au maximum **P2**.

# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

**Les associations :** une association représente une relation structurelle entre classes d'objets. La plupart des associations sont binaires, c'est à dire qu'elles connectent deux classes. On représente une association en traçant une ligne entre les classes associées.

## Agrégation

Une association particulière non symétrique. Il s'agit d'une relation entre deux classes, spécifiant que les objets d'une classe (classe agrégat) sont des composants de l'autre classe (classe agrégée). Une relation d'agrégation permet donc de définir des objets composés d'autres objets.

L'agrégation est représentée par un losange clair associé à l'agrégat.

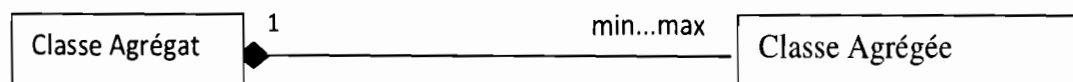


## Composition

C'est une agrégation forte. Elle exprime une relation de contenance.

Dans une composition, la classe agrégat est responsable de la mise à disposition de ses parties. La suppression d'un objet agrégat entraîne la suppression des objets agrégés. La valeur maximale de multiplicité du conteneur ne doit pas excéder 1 car les objets, instances de la classe des composants, appartiennent au même objet conteneur.

La composition est représentée par un losange noir.

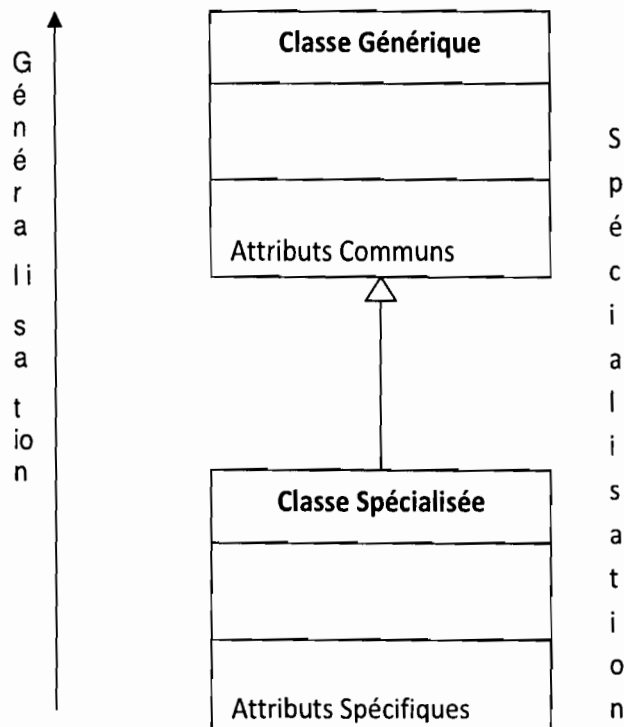


# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

## Généralisation

UML emploie le terme de généralisation pour désigner la relation de classification entre un élément plus général et un élément plus spécifique. La relation de généralisation signifie « est un » ou « est une sorte de ».

La classe générique porte les attributs communs à toutes les classes spécifiques. Mais une classe spécifique ne porte que les attributs spécifiques à son type de classe.



# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique

- **Syntaxe**

**Attributs** : la syntaxe générale est :

[Visibilité] nom attribut [multiplicité]: type [=valeur initiale]

La multiplicité est le nombre d'occurrences possibles de l'attribut.

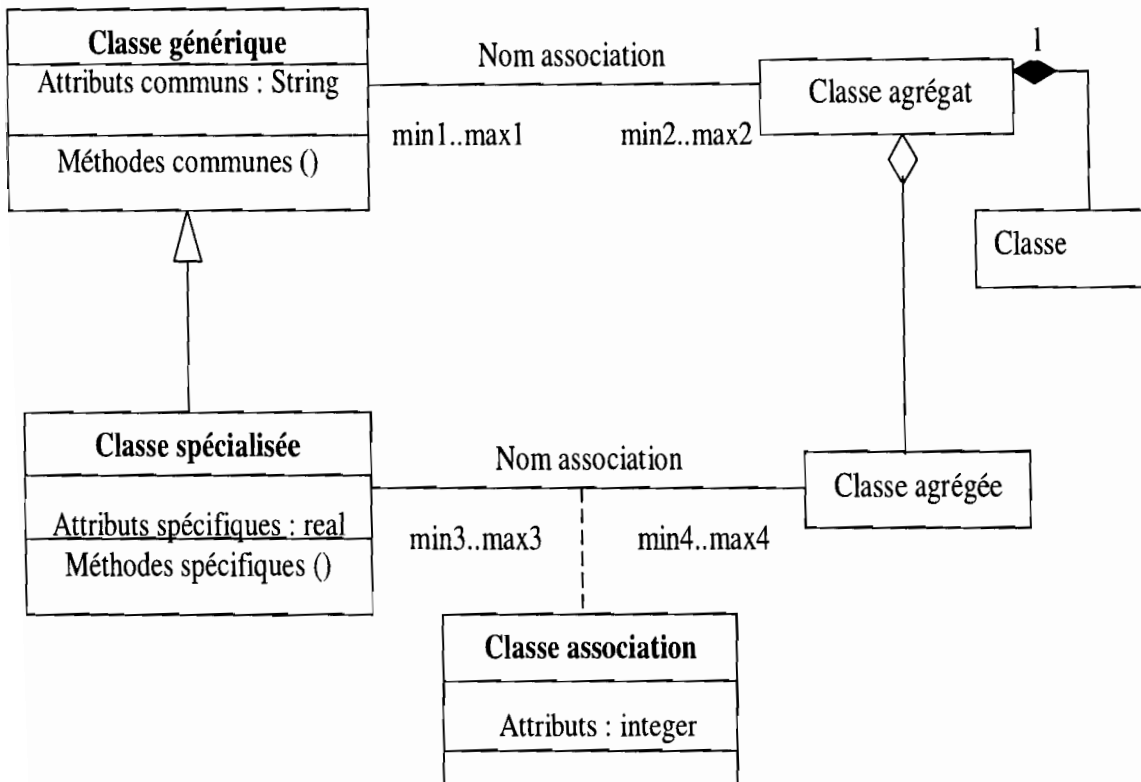
**Méthodes** : la syntaxe est :

Visibilité Nom (liste paramètre) type {propriétés}

Liste paramètre est représentée par : Nature Nom : type =Valeur par défaut

La nature est soit, **In**, soit **Out** ou encore **InOut**.

➤ **Formalisme utilisé**



# Mémoire de fin de Cycle d'Ingénieur de Conception Informatique