Université Sidi Mohamed Ben Abdellah Faculté des Sciences et Techniques Fès Département Informatique



Gestion de distribution d'eau potable

Association El BOUGRINIA de l'eau potable

Rapport de Stage

Réalisé par :

M. Hicham EL GUAROUANI

Supervisé par :

M. Azeddine ZAHI

REMERCIEMENTS

Une agréable occasion se présente, enfin à moi, pour exprimer ma reconnaissance auprès de toutes les personnes, dont l'intervention a contribué à la réalisation de ce travail.

Je tiens à remercier **Mr. Mohamed BOUAÂLAM**, Président de l'**ABEP**, pour l'opportunité qu'il m'a octroyée en acceptant de me prendre en stage au sein de l'association et aussi pour son encadrement à l'ABEP.

Aussi, j'aimerai exprimer ma profonde gratitude au **Pr. Azeddine Zahi** pour son encadrement exceptionnel, ses directives précieuses, ses conseils avisés, et son soutient remarquable lors des réunions qu'il m'a prodigué tout au long de ce stage.

Mes vifs remerciements vont aux membres du jury, **Pr. L. LAMRINI, Pr. F. LAMRABET, Pr. A. ZAHI**, d'avoir accepté de juger mon travail.

Que le corps professoral du département Informatique de la FSTF trouve ici mes vifs remerciements pour la qualité de l'enseignement qu'ils nous ont offert.

TABLE DES MATIERES

Introduction	5
CHAPITRE 1 : PRESENTATION DU PROJET	7
1. Organisme d'accueil : Association AL BOUGRINIA	7
1.1. Présentation de l'association	7
1.2. Organisation de l'association	8
2.Description du projet	8
2.1. Objectifs	8
2.2. Contraintes	9
2.3. Solution proposée	10
3. Démarche suivi : Processus 2TUP	10
CHAPITRE 2 : ETUDE PRELIMINAIRE DU PROJET	13
1.Cahier des charges	13
2. Modélisations du contexte	18
2.1. Identification des acteurs	18
2.2. Identification des messages	18
CHAPITRE 3 : ETUDE FONCTIONNELLE	23
1.Capture des besoins fonctionnels	23
1.1. Identification des cas d'utilisation	23
1.2. Description détaillée des cas d'utilisation	28
2. Analyse	32
2.1. Modèle statique	32
2.2. Modèle dynamique	35
CHAPITRE 4: ETUDE TECHNIQUE	40
1. Architecture et organisation en couches	40
2. Conception Générique	41
3. Outils de réalisation	45
Chapitre 5: Realisation	47
CONCUISION	<i>/</i> /8

Rapport de Stage - Gestion de distribution d'eau potable

INTRODUCTION

Dans le cadre de la formation, à la **Faculté des Sciences et Techniques de Fès,** nous sommes amenés à réaliser un stage de fin d'études, afin de compléter les connaissances théoriques acquises et de mieux les appliquer dans le monde du travail.

Notre stage a été effectué à *l'Association Al-Bougrinia de l'eau potable* de la commune Ain Timganay, Cercle El Menzel Province de Séfrou. Etant consciente de l'importance et de l'apport considérable des systèmes d'information dans le processus de gestion, l'association nous a confié la tâche de développer un système d'information pour la gestion de ses activités.

Le présent document constitue le rapport du projet réalisé lors de ce stage, il est divisé en 5 chapitres.

Le premier définit le cadre général de notre projet, il présente en premier lieu l'organisme d'accueil, puis une description du projet ainsi que la démarche suivi pour mener à bien ce projet.

Le deuxième chapitre présente l'étude du projet, il décrit le cahier des charges à réaliser ainsi que le contexte du système à développer.

Le troisième chapitre est consacré à l'étude fonctionnelle qui s'intéresse à la spécification des besoins et à l'analyse du système.

Le chapitre quatre donne une description de l'étude technique, il présente l'architecture de notre système d'une part et les outils de développement utilisés d'autres parts.

Le dernier chapitre est dédié à la présentation du système que nous avons réalisé.

Chapitre 1 : Présentation du projet

Dans ce premier chapitre nous présenterons l'organisme d'accueil, une description générale de notre projet, ainsi que la démarche suivie.

CHAPITRE 1: PRESENTATION DU PROJET

1. Organisme d'accueil : Association AL BOUGRINIA

1.1. Présentation de l'association

L'organisme concerné par notre projet de fin d'études est « l'Association AL-Bougrinia de l'eau potable ». C'est une association qui a été créée, en 19xx à la commune Ain Timganay, cercle EL Menzel, province de Séfrou, par les habitants. L'objectif primaire de cette association est la gestion de la distribution de l'eau potable dans la commune Timganay. AL-Bougrinia compte, actuellement, un réseau de distribution d'eau potable couvrant le territoire de la commune et des milliers d'adhérents. Les adhérents sont soit des particuliers (habitants de la commune), soit des institutions telles que l'école primaire, le collège, le dispensaire et la commune.

La mission d'AL-Bougrinia consiste à accomplir et veiller sur le bon déroulement des activités suivantes :

- Gestion des contrats d'adhésion. En effet, pour bénéficier des services de l'association en matière d'eau potable, l'intéressé doit signer un contrat d'adhésion avec l'association.
- Perception des cotisations couvrant la consommation relevée sur les compteurs des adhérents.
- Gestion et maintenance du château et du réseau de distribution d'eau installés sur le territoire de la commune.
- Installation des points de distribution d'eau et des compteurs chez les adhérents de l'association.
- Restitution des compteurs déjà installés en cas panne ou en cas de résiliation demandée par l'adhérent ou forcée par l'association.

1.2. Organisation de l'association

L'association est gérée par un bureau, élu par les adhérents, dont les membres sont des bénévoles. Le bureau est constitué d'un président, d'un vice-président, d'un trésorier, et des conseillers. L'association dispose, également, d'un certain nombre d'employés permanents embauchés pour assurer les activités quotidiennes de l'association :

- Un administrateur (Agent commercial) qui s'occupe de la gestion commerciale: adhésions, résiliations, règlement des cotisations, comptabilité, etc.
- Les techniciens qui assurent les activités techniques telles que l'installation et la restitution des compteurs, le relevé des consommations, le traitement des incidents etc.
- Le magasinier qui se charge de la gestion du stock d'équipements hydrauliques utilisés dans l'installation et la maintenance du réseau de distribution.

Les employés permanents de l'association sont supervisés par le président de l'association.

2. Description du projet

2.1. Objectifs

Actuellement, toutes les activités de gestion au sein de l'association sont effectuées manuellement, en se basant uniquement sur le support papier. Ceci rend le suivi du travail, par les membres du bureau, une tâche difficile, complexe et non fiable. Pour cela, le bureau de l'association a décidé de mettre en place un système d'information pour la gestion de toutes les activités. Le système future doit permettre aux employés de l'association de réaliser rigoureusement, correctement et aisément les différentes tâches qui leurs sont confiées, à savoir :

- Gestion des contrats d'adhésion
- Gestion des compteurs
- Gestion des cotisations
- Gestion des crédits
- Gestion de la comptabilité
- Gestion du stock
- Gestions des incidents

2.2. Contraintes

Le système à développer doit prendre en considération les contraintes suivantes :

- Capacité à supporter la volumétrie des données de plus en plus croissante. En effet, l'association gère environ 1000 contrats, ceci induit au moins 2000 enregistrements par mois, concernant les consommations et les débits financiers. En plus, les transactions qui peuvent avoir lieu tels que les nouveaux contrats, les résiliations, les incidents, etc., doivent aussi être enregistré.
- Evolutivité dans le sens où le système doit être facilement extensible à de nouvelles fonctionnalités telles que la gestion graphique et interactive des incidents, la saisie des consommations en temps réel, la gestion des employés, etc.
- Sécurité des données dans le sens où le système ne doit permettre l'accès à certaines fonctionnalités qu'aux personnes disposant des privilèges correspondants.
- Accessibilité: le système doit permettre aux membres du bureau de suivre l'évolution des activités au sein de l'association depuis leurs lieux de travail. En effet, on doit prendre en compte le fait que les membres du bureau sont des bénévoles et ne se trouvent pas forcement sur les locaux de l'association.

2.3. Solution proposée

Pour répondre aux objectifs et contraintes précédemment annoncés, nous avons proposé une solution Client-serveur avec deux volets:

- Le premier de type client lourd s'appuyant sur une architecture 2tiers. Ce volet sera destiné aux employés qui accomplissent leurs tâches dans le local de l'association
- Le deuxième de type client léger basé sur une architecture 3-tiers.
 Ce volet sera essentiellement destiné aux membres du bureau en déplacement permanent.

3. Démarche suivi : Processus 2TUP

Pour la réalisation du projet, nous avons opté pour le processus 2TUP « 2 Track Unified Process», ou simplement le processus en Y, comme processus de développement. Ce processus dissocie les aspects fonctionnels des aspects techniques permettant ainsi de répondre aux contraintes de changement continuel imposées au système. Il s'articule sur 3 branches (figure 1):

- La branche gauche (fonctionnelle): Capitalise la connaissance du métier de l'entreprise indépendamment des technologies utilisées.
 Cette branche comporte les étapes suivantes :
 - La capture des besoins fonctionnels qui produit, un modèle des besoins qui se focalise sur le métier de l'entreprise.
 - L'analyse qui consiste à étudier précisément la spécification fonctionnelle du système future en termes de métier.
- La branche droite (architecture technique) : capitalise un savoirfaire technique pour le système indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

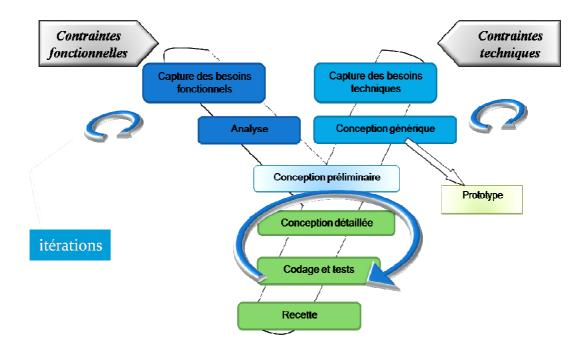


Figure 1: Processus de développement 2TUP (en Y)

- La capture des besoins techniques qui spécifie les contraintes,
 les besoins non fonctionnels et les choix conditionnant la conception du système.
- La conception générique consiste à construire l'architecture technique du système qui doit être la moins dépendante possible des aspects fonctionnels. Cela revient à construire le squelette du système en décrivant les composants nécessaires et leurs interactions. Cette étape se concrétise par la production d'un prototype.
- La branche du milieu : à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette branche comporte les étapes suivantes :
 - La conception préliminaire qui consiste à intégrer le modèle d'analyse dans l'architecture technique.
 - La conception détaillée qui définit comment réaliser chaque composant.
 - Le codage et les tests des composants.

Chapitre 2 : Etude préliminaire du projet

Lors de ce chapitre nous mènerons l'étude préliminaire de notre projet. Nous présenterons précisément le cahier des charges obtenu ainsi que la modélisation du contexte du système futur.

CHAPITRE 2: ETUDE PRELIMINAIRE DU PROJET

1. Cahier des charges

L'élaboration du cahier des charges de notre projet s'est déroulée en deux étapes :

- Etude, critique et complétion d'un cahier des charges préliminaire,
 réalisé dans le cadre d'un autre stage.
- Validation du cahier des charges final lors des réunions tenues, dans l'association, avec les employés, le président de l'association et mon encadrent à la FST.

Nous avons divisé le cahier des charges en plusieurs parties, chacune représente un aspect à gérer par le système.

Gestion des contrats d'adhésions : Le système doit permettre à l'agent commercial :

- D'enregistrer les demandes d'adhésion. En effet, pour bénéficier des services de l'association, les concernés doivent faire une demande d'adhésion auprès de l'administrateur. Cette opération se termine par l'impression du contrat d'adhésion, en deux exemplaires, au nom de l'intéressé pour légalisation.
- De valider les demandes d'adhésion. Une demande d'adhésion est validée après la présentation des contrats légalisés et le règlement des frais d'adhésions, qui sont fixés à 1050DH. Un reçu et un ordre de mise en place du compteur sont édités suite à cette opération. La demande d'adhésion d'un adhérent infidèle (renouvellement du contrat) ne sera validée qu'après le paiement des montants dus et d'une amande de 150 DH. Les frais d'adhésion peuvent être réglés

dans leur globalité ou en partie sous-forme de crédit. Dans ce dernier cas un est imprimé pour légalisation.

- De rejeter une demande dans le cas d'inéligibilité ou d'abandon.
- De consulter la fiche d'un adhérent et aussi de modifier ses informations en cas d'erreur, mais avant la validation du contrat.
- D'enregistrer une résiliation d'un contrat à la demande de l'adhérent. La résiliation n'est validée qu'après le règlement des montants dus. Un ordre de restitution du compteur est édité suite à cette opération.
- D'enregistrer une résiliation d'un contrat par l'association pour les adhérents infidèles. Un adhérent est qualifié d'infidèle s'il a trois factures impayées. Un avis d'avertissement est édité puis envoyé au concerné avant de procéder à la résiliation. Cette dernière n'est pas conditionnée par le règlement des montants dus.
- D'éditer, mensuellement, la liste des adhérents pour les quelles une résiliation forcée doit avoir lieu.
- D'éditer les ordres de résiliations forcées.

Gestion des compteurs : Il existe deux types de compteurs à gérer : (1) Les compteurs associés à un contrat d'adhèrent, (2) les compteurs associés à un secteur, appelés contrôleurs. Ces derniers permettent de contrôler la consommation et d'identifier les éventuelles fuites au niveau des secteurs. Ainsi, la différence de consommation relevée est comptabilisée sur les adhérents du secteur de manière proportionnelle. Pour les deux types le système doit permettre à l'agent commercial :

- D'enregistrer un compteur installé. En effet, après l'installation d'un compteur, le technicien retourne l'ordre de mise en place, signé par l'adhérent concerné, à l'administrateur pour enregistrement. Le compteur est soit remis à zéro, soit enregistré avec l'ancien index.
- D'enregistrer la restitution d'un compteur. En effet, le technicien retourne l'ordre de restitution à l'administrateur pour enregistrement.

- D'enregistrer, mensuellement les index des compteurs d'adhérents et ceux des contrôleurs. Au 25 de chaque mois les techniciens recueillent les nouvelles consommations enregistrées au niveau de tous les compteurs installés.
- De consulter les informations sur les compteurs.
- D'éditer les consommations globales de l'association avec mention des périodes où la consommation est basse ou élevée.

Gestion des cotisations : Le système doit permettre :

D'éditer mensuellement et de manière automatique les avis de payement de cotisations pour tous les adhérents. Les paiements en retard sont signalés sur l'avis de paiement en ajoutant un montant additionnel de 5dhs par facture. Le montant de cotisation à payer est calculé par tranches à partir de la consommation relevés sur le compteur de l'adhérent et du contrôleur de son compteur. La consommation donnée par la formule :

$$Consommation = (indCot - AncIndCot) \times \frac{(indCtr - AncIndCtr)}{\sum (indCot - AncIndCot)}$$

- D'éditer la facture, après chaque règlementation.
- D'éditer, mensuellement, la liste des adhérents retardataires. C'està-dire ayant des factures impayées et qui sont appelés à régulariser leurs situations.
- D'éditer, mensuellement, les avis d'avertissement pour les envoyer aux adhérents retardataires.

Gestion des crédits: L'association offre la possibilité à ses adhérents de bénéficier d'un prêt sans intérêts, sur une partie ou la totalité des frais d'adhésion et aussi sur les montants dus dans le cas des consommations jugées lourdes. Le crédit est attribué selon l'éligibilité de l'adhérent et remboursé selon un mode de versement mensuel. L'association offre aussi la possibilité à ses adhérents; ceux qui ne sont disponible qu'à une période de l'année; de régler leur cotisation d'avance. Dans ce cas le règlement sera considéré comme un crédit négatif, c'est-à-dire en faveur de l'adhérent. Un avoir est délivré à l'adhérent suite au règlement. La mêmes procédure est appliquée dans le cas d'erreur; c'est-à-dire l'adhérents paye un montant supérieur à sa consommation. Pour tout ceci le système doit nous permettre:

- D'enregistrer les demandes de crédit.
- De valider les demandes de crédit.
- D'enregistrer, mensuellement, les opérations de paie, ces dernières seront intégrés aux avis de paiements, et de délivrer par la suite un avoir à l'adhérent.

Gestion de la comptabilité : Ce volet sera essentiellement destiné au membre du bureau de l'association pour suivre le flux financier de l'association. Dans ce cadre le système doit permettre :

- D'enregistrer les dépenses, généralement lors d'achat de matériel et des travaux de réparation.
- D'enregistrer les recettes. Les recettes sont principalement issues des montants des factures.
- D'éditer le bilan de l'association, permet d'établir un bulletin annuel ou semestriel du flux financier de l'Association.

Gestion du stock: Le stock de l'association comprend essentiellement les équipements hydromécaniques nécessaires à l'installation et la maintenance du réseau de distribution tels que les conduites, les compteurs, les raccords, etc. Le système doit permettre:

- D'enregistrer les sorties du stock, le cas des compteurs mis en place et des articles utilisés lors des travaux de maintenance ou d'extension des canalisations.
- D'enregistrer l'entrée des nouveaux articles avec leurs quantités.
- D'éditer les articles en rupture de stock.
- De préparer les commandes des articles ayant atteint le seuil du stock-alerte.
- De préparer des commandes urgentes des articles en rupture du stock.
- D'éditer les articles les plus consommés par l'association, avec mention des fournisseurs.

Gestion des Incidents : Suite à un incident, le technicien se déplace sur les lieux pour faire le diagnostic et identifier les dépenses et les articles nécessaires pour la réparation de la panne. Le système doit permettre :

- D'enregistrer les incidents avec mention des matériels endommagés.
- D'éditer un bon de réparation contenant le matériel nécessaire,
 coût et le sujet de la panne (réseau, adhérent,...).
- D'éditer les incidents avec mention des périodes où les interventions sont élevées.
- D'établir un bilan sur les secteurs défectueux (où se produisent les pannes constamment), et les périodes rouges.

2. Modélisations du contexte

2.1. Identification des acteurs

Un acteur système peut faire abstraction d'un groupe de personnes effectuant le même rôle du point de vue du système. Toutefois une seule personne physique peut jouer plusieurs rôle vu du système dans ce cas un acteur sera considéré pour chaque rôle joué. Le futur système disposera de 5 acteurs principaux repartis de la manière suivante :

- L'agent commercial: représente l'acteur principale du système et dispose de la plupart des droits concernant la gestion des différents volets du système: gestion des contrats, des compteurs, des cotisations et des crédits.
- Le magasinier: se charge essentiellement de la gestion du stock et de ses entrées et sorties.
- Le technicien: c'est l'acteur en charge du maintien des incidents et gère ces dernières au niveau du système.
- Le membre du bureau : ce dernier est concerné par le système dans sa globalité et dispose des privilèges nécessaires pour la consultation et l'édition de tous les messages et situation produites par l'application. Il peut ainsi établir les bilans périodiques concernant les adhérents, les cotisations, la comptabilité, les crédits, le stock et les incidents.
- Administrateur: gère les profils des différents types d'utilisateurs du système. Il effectue aussi les opérations techniques au niveau du système tels que le backup.

2.2. Identification des messages

Un message représente tout moyen de communication unidirectionnelle entre les objets qui transportent de l'information et qui déclenche une activité chez le récepteur par la suite. Nous avons parvenu à identifier un nombre de messages échangés entre le système et les différents acteurs :

Les messages émis par le système

Gestion des contrats

- Les contrats d'adhésion, délivrés en vue de légalisation suite aux demande d'adhésion auprès du l'agent commerciale.
- Les reçus d'adhésion, suite à la validation des demandes et après règlement des frais d'adhésion.
- La liste des adhérents de l'association.

Gestion des cotisations

- Les avis de paiement (mensuellement).
- Les factures.
- Les avis d'avertissement (suite à deux mois de retard).
- La liste des adhérents retardataires.
- La liste des adhérents en vue de résiliation.
- Les avis de résiliation (suite à plus de deux mois de retard et après validation de la part de l'agent commerciale).

Gestion des compteurs

- Les ordres de mise en place des compteurs.
- Les ordres de restitution des compteurs (suite à une résiliation).
- La liste des compteurs (avec toutes les informations).
- Bilans périodiques de la consommation.

Gestion de la comptabilité

- La liste de dépenses de l'association.
- La liste de recettes de l'association.
- Bilans périodique du flux financier.

Gestion des crédits

Les contrats de crédits.

- Les avoirs tenant au courant les adhérents de l'état de leurs crédits.
- La liste des crédits (avec classification).

Gestion du stock

- La liste des articles du stock (avec mention des états).
- Les bons de commande pour les articles ayant atteint le seuil du stock-alerte.
- Les bons de commande urgents pour les articles en rupture du stock.
- Bilans de l'état du stock (avec mention des articles, leurs qualités, leurs fréquences d'utilisation et leurs fournisseurs).

Gestion des incidents

- Les ordres d'interventions (après déclaration d'incidents).
- Les bons de réparation avec les charges (matériel et coût) estimés nécessaire pour la panne.
- La liste des incidents et leurs états (avec classification).
- Bilans des secteurs et périodes défectueuses.

Gestion des profils

La liste des profils utilisateurs.

Les messages reçus par le système

Gestion de contrats

- Ajout d'adhérents.
- Validation de demande d'adhésion.
- Modification d'adhérents.
- Résiliation et rejet de demandes.

Gestion des cotisations

Modification des avis de paiement.

Validation résiliation (suite à une infidélité ou à la demande).

Gestion des compteurs

- Mise à jour des index.
- Mise en place de compteurs.

Gestion de la comptabilité

- Ajout de dépense.
- Ajout de recette.
- Etablissement de bilan financier.

Gestion des crédits

- Demande de crédit.
- Validation de demande de crédits.

Gestion du stock

- Ajout de nouvelles entrées (nouveaux articles).
- Modification des quantités (inclus les entrées et les sorties).

Gestion des incidents

- Enregistrement d'un nouvel incident.
- Modification de l'état d'un incident (après intervention).
- Enregistrement des charges d'un incident.

Gestion des profils

- Authentification.
- Ajout de nouveaux profils.
- Modification des profils.
- Suppression des profils.

Chapitre 3: Etude Fonctionnelle

Dans ce chapitre nous traiterons le coté fonctionnel du système grâce au concept des cas d'utilisation. Nous donnerons une description détaillée des cas d'utilisation, le modèle statique ainsi que le modèle dynamique.

CHAPITRE 3: ETUDE FONCTIONNELLE

1. Capture des besoins fonctionnels

1.1. Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable et intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. La figure 2 montre le diagramme des cas d'utilisation global.

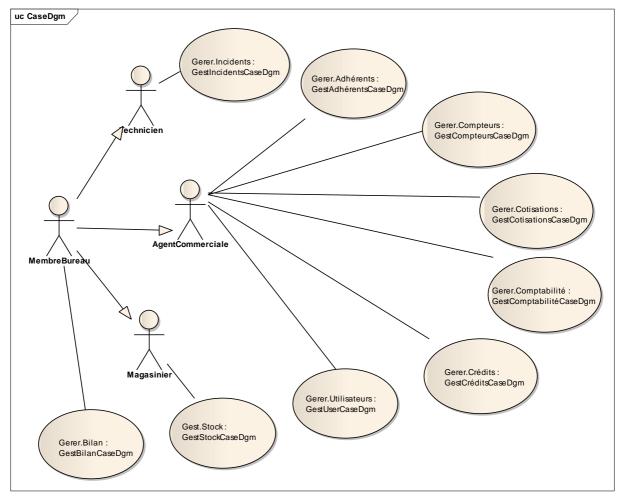


Figure 2 : Diagramme de cas d'utilisation général

Dans la même vision de modularité nous avons répartis les cas d'utilisation identifiés en gestions.

Gérer les Contrats

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
Agent	Enregistrer demande d'adhésion	Reçoit: - Les informations de l'adhérent. Émet: - Contrat à légaliser.
	Valider Demande	Reçoit: - Numéro du contrat d'adhésion. Émet: - Liste de contrats en attente de validation Reçu d'adhésion.
	Éditer adhérent	Reçoit: - Numéro du contrat d'adhésion Nouvelles informations. Émet: - Les informations de l'adhérent.
	Résilier Contrat	Reçoit: - Numéro du contrat d'adhésion. Émet: - Avis de résiliation - Liste des contrats résiliables.
	Éditer résiliation	Reçoit: - Numéro du contrat d'adhésion. Émet: - Liste de contrats résiliés. - Reçu de réglementation.

Tableau 1 : Identification des cas d'utilisations 1 (gérer les adhérents)

Gérer les compteurs :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
Agent commercial	Enregistrer compteurs	Reçoit: - Les informations du compteur (numéro du compteur, référence d'adhérent,). - Index initial du compteur. Émet: - Ordre de mise en place.

Restituer compteur	Reçoit: - Numéro du compteur. Émet: - Liste des compteurs correspondants aux contrats résiliés. - Ordre de restitution.
Éditer compteur	Reçoit: - Numéro du compteur Nouvel index Nouvelle référence d'adhérent. Émet: - Les informations du compteur Nouvelle consommation.
Etablir bilan	Reçoit: - Date début et date fin Numéro du compteur. Émet: - Consommations.

Tableau 2: Identification des cas d'utilisations 2 (gérer les compteurs)

Gérer les cotisations :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
Agent commercial	Éditer les avis de paiement	Reçoit: - Numéro du compteur. Émet: - Avis de paiement.
	Éditer Les cotisations	Reçoit: - Numéro du compteur. - Montants de cotisation. Émet: - Reçu.
	Éditer Cotisations retardées	Reçoit: - Numéro du compteur. Émet: - Liste des adhérents retardataires Liste de toutes les cotisations retardées.
	Etablir bilan	Reçoit: - Date début et date fin. Émet: - Recettes.

Tableau 3: Identification des cas d'utilisations 3 (gérer les cotisations)

Gérer les crédits :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
,	Enregistrer demande de crédit	Reçoit: - Numéro du contrat d'adhésion Type de crédit. Émet: - Contrat du crédit.
	Valider demande de crédit	Reçoit: - Numéro du contrat d'adhésion. Émet: - Reçu de crédit.
Agent commercial	Éditer Les crédits	Reçoit: - Numéro du contrat d'adhésion Changement du crédit (montant a retranché). Émet: - Avoir sur l'état du crédit Liste des crédits avec informations.
	Etablir bilan	Reçoit: - Numéro du contrat d'adhésion. - Date début et date fin. Émet: - Crédits.

Tableau 4: Identification des cas d'utilisations 4 (gérer les crédits)

Gérer la comptabilité :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
	Enregistrer recette	Reçoit: - Montants encaissé. Émet: - Reçu de recette.
Agent commercial	Enregistrer dépense	Reçoit: - Informations Montant dépensé. Émet: - Reçu de dépense.
	Etablir bilan	Reçoit: - Date début et date fin. Émet:

	Flux financiers.Bilan sur l'état financier de l'association.	
--	---	--

Tableau 5: Identification des cas d'utilisations 5 (gérer la comptabilité)

Gérer Le stock :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
Magasinier	Enregistrer sortie	Reçoit: - Reference d'article Quantité de sortie. Émet: - Bon de sortie.
	Enregistrer entrée	Reçoit: - Reference d'article Quantité d'entrée. Émet: - Etat de l'article dans le stock Bon d'entrée.
	Préparer commande	Reçoit: - Reference d'article Quantité de la commande. Émet: - Bon de commande.
	Etablir bilan	Reçoit: - Critère de classification (Marque, fournisseur,). Émet: - Statistique selon divers critères Bilan sur l'état du stock.

Tableau 6: Identification des cas d'utilisations 6 (gérer le stock)

Gérer Les incidents :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
Technicien	Enregistrer incident	Reçoit: - Information incident charges nécessaires à la réparation. Émet: - Ordre d'intervention.
	Éditer incident	Reçoit: - Numéro d'incidents informations.

	 Nouvel état (après intervention). Émet: Liste des incidents.
Etablir bilan	Reçoit: - Critère de classification (période, endroit,). Émet: - Statistique selon divers critères. - Bilan sur l'état du réseau.

Tableau 7: Identification des cas d'utilisations 7 (gérer les incidents)

Gérer Les profils utilisateurs :

Acteur principal	Nom du cas d'utilisation	Messages émis/reçus par le système
Admin	Enregistrer utilisateur	Reçoit: - Informations utilisateur. Émet: - Confirmation d'inscription.
	Éditer utilisateur	Reçoit: - Identifiant utilisateur informations. Émet: - Liste des utilisateurs.
	S'authentifier	Reçoit: - Identifiant, mot de passe. Émet: - Etat du système (selon les privilèges de chaque utilisateur).

Tableau 8: Identification des cas d'utilisations 8 (gérer les profils)

1.2. Description détaillée des cas d'utilisation

La description détaillée d'un cas d'utilisation consiste à définir clairement le début et la fin; recenser et ordonner, de façon textuelle, toutes les interactions et les messages échangés entre les acteurs et système. Les interactions sont organisés dans des **scénarios**, un scénario représente une succession particulière d'enchaînements et un enchainement est une

séquence d'interactions. Il faut également préciser les variantes possibles à savoir les cas nominaux, les cas alternatifs et les cas d'exceptions.

Dans cette section nous n'allons présenter que la description détaillée du cas d'utilisation « **gestion des cotisations** » les autres cas d'utilisation seront présenté dans le dossier d'analyse (annexe n°1). Ainsi, le long de ce chapitre nous ne détaillerons que ce cas d'utilisation. La figure 3 montre le diagramme des cas d'utilisation « gestion des Cotisations ».

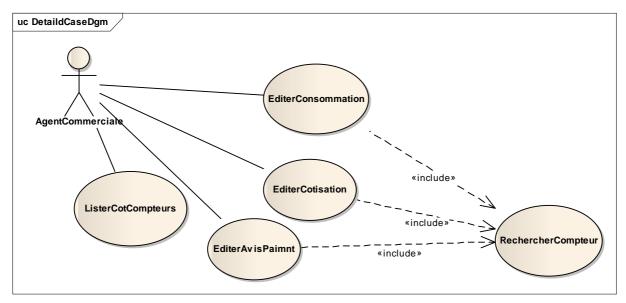


Figure 3 : Diagramme de cas d'utilisation (Gerer les cotisations)

Sommaire d'identification

Titre: Gérer les Consommations

But : éditer la consommation des compteurs.

Résumé: éditer mensuellement la consommation marquée auprès de l'ensemble des compteurs.

Acteur: Agent commercial

Description des enchainements

Pré-conditions: L'agent commercial est authentifié

Scenario nominale:

Ce cas d'utilisation commence lorsque L'agent commercial demande au système :

- d'Editer la consommation chaque 25 du mois

Enchainement (a): Enregistrer nouvelles consommations

Enregistrer mensuellement la nouvelle consommation, en incrémentant automatiquement le mois vers en cours et l'agent commercial introduit un nouvel index.

Enchainement alternatifs:

Enchainement (b): Editer consommations

Editer selon le choix l'index d'un compteur déjà saisi après notification d'erreur auprès de l'adhérent

Exceptions:

[Exception 2 : Index Erreur] : Index précédant supérieur à l'index en cour.

Post-conditions:

- 1- Consommation éditée avec succès.
- 2- Génération des avis de paiement.

Besoins IHM:

Pour modifier l'index d'un compteur l'agent commercial dois répertorier la liste des compteurs et effectuer la saisi directement vu que la tache se répète un grand nombre de fois « > Compteurs x nombre d'opération ».

Sommaire d'identification

Titre: Gérer les cotisations

But : Editer les cotisations des adhérents.

Résumé: Editer automatiquement les avis de cotisations des adhérents, éditer les cotisations et les cotisations retardées.

Acteur: Agent commercial

Description des enchainements

Pré-conditions:

- 1- L'agent commercial est authentifié
- 2- Au moins un compteur enregistré
- 3- Nouvelles consommations enregistrées (après 25 de chaque mois)

Scenario nominale:

Ce cas d'utilisation commence lorsqu'il y a :

- nouvelles consommations enregistrées (après 25 de chaque mois).
- réglementation des cotisations retardées.

Enchainement (a): Editer les avis de paiement

Éditer et générer automatique de façon incrémental l'ensemble des avis de paiement après l'enregistrement de chaque nouvelle consommation (après calcule de la consommation mensuelle et du montant de la cotisation), après le 25 de chaque mois.

Scenarios:

- 1- Auto-incrémentions des numéros des compteurs.
- 2- Extraction des deux derniers index du compteur (à partir de l'index compteur et de l'index de la dernière cotisation).
- 3- Calculer consommation en ajoutant la valeur calculée par les contrôleurs répartis entre les compteurs du même secteur.
- 4- Répartir en tranches.
- 5- Extraire les tarifs selon les tranches.
- 6- Ajouter la part mensuelle de crédit (s'il existe).
- 7- Calculer montant net à payer.

Enchainement (b): Editer cotisation

Au règlement des montants de cotisation l'agent commercial édit ces dernières manuellement en un état ''réglé'' ou ''payée'' et livre un reçus de cotisation pour chaque instance réglée. Ajouter une amande en cas de cotisation retardée.

Exceptions:

[Exception 1 : Compteur Inexistant] : numéro du compteur à éditer n'existe pas

[Exception 2 : Index Erreur] : Index précédant supérieur a l'index en cour.

Post-conditions:

- 1- Tous les avis généré => impression des avis.
- 2- Cotisation édité avec succès => impression d'un reçu de cotisation.

Besoins IHM:

- L'édition des avis de cotisation dois se faire de façon automatique, l'agent commercial ne doit intervenir qu'au lancement du processus ou de le suspendre et reprendre en cas d'erreur.
- L'édition des cotisations nécessite une recherche par numéro de contrat.

2. Analyse

2.1. Modèle statique

A partir de l'étude préliminaire et de la description détaillée des cas d'utilisation nous avons élaboré le diagramme de classes présenté dans la figure 4.

- La classe Adhérents : permet de stocker les informations nécessaires à l'identification des adhérents.
- La classe Contrat :Chez l'association il existe des adhérents avec plus qu'une seule adhésion chose qui a nécessiter de gérer les contrats séparément des adhérents tout en gardant contact avec ces derniers dans le cadre d'une relation d'agrégation multiple (un adhérent seul peut avoir un contrat ou plus). Cette classe s'occupera de la gestion des différents états des contrats a savoir non-valide, valide et résiliée.
- La classe compteur: Nous permettra de gérer les compteurs dans l'ensemble de leurs états (en stock, actif et restitué). La classe compteur sera consacré à l'enregistrement, restitution et édition des propriétés des compteurs.

- La classe contrôleur: Les contrôleurs sont des compteurs qui appartiennent à l'association et qui servent à contrôler la consommation au niveau des secteurs et d'équilibrer les consommations des cotisations suite à une formule d'équilibrage pour éviter à l'association toute perte éventuelle due aux fuites ou aux fractions. Cette classe sera similaire a la précédente mais sans relation direct avec un contrat.
- La classe secteur: A chaque secteur appartiens un certain nombre de compteurs et c'est ces secteurs qui nous permettront une classification de ces compteurs pour faciliter une saisie de consommation appropriée. Permettra d'attribuer un compteur à un secteur et de contrôler les secteurs par la suite.
- La classe cotisation : Enregistrera une instanciation par mois pour tout mois d'adhésion, marquant la consommation ainsi que tous les dues mensuels des adhérents. Concerné en principe par l'enregistrement des consommations et l'élaboration des avis de paiement.
- La classe control: Générer après chaque saisi des index des contrôleurs dans un but de garder un historique pour toute éventuelle vérification dans le futur. Par la suite cette classe sera utilisée dans la formule d'équilibrage mis au point pour déterminer la vraie valeur de consommation.
- La classe crédit: Pour chaque crédit créer on doit enregistrer un objet crédit, à spécifier qu'un même contrat peut disposer de plus qu'un crédit. Pour l'ajout modification et réglementation des crédits. Cette classe généralise le cas de trois autre classe représentant chacune un type des modes de payement proposé par l'association (3, 6 et 12 mois).
- La classe versement: Pour enregistrer les versements des parts de crédit mensuellement. Pour enregistrer un versement d'un crédit quelconque.

- La classe débit : Pour gérer le flux financier de l'association il faudra enregistrer tout genre de débits que ce soit dépenses ou recettes, c'est pourquoi on opte pour une classe débit englobant les deux cas. Enregistrer un échange financier positif (recette) ou négatif (dépense).
- La classe incident: Nous permettra de garder un œil sur l'état du réseau et de la canalisation, en sauvegardant chaque incident à travers une instance de cette classe. Enregistrer un incident et éditer son état ainsi qu'en enregistrer les charges.
- La classe Article: Pour gérer le stock et cela en associant une instance article à chaque article dans notre stock. Enregistrer un article, éditer ses informations
- La classe commande: Pour commander les articles ayant atteint le seuil du stock alerte.
- La classe débit-Stock: Gere les variations au niveau du stock, essentiellement utilisé pour sauvegarder toutes les opérations de débits effectuées sur le stock, dues aux incidents ou à l'utilisation du matériel.
- La classe tarif: Pour donner un aspect générique à l'application on a opté pour une entité tarif qui contiendra tous les genres de tarifs ainsi que leurs désignations et les montants dues à ces transactions.
- La classe utilisateur: Elaboré principalement pour la gestion des opérations sur les utilisateurs (ajout, modification et Authentification).
 Mais aussi utilisé pour la détermination des droits de chaque catégorie d'utilisateur.

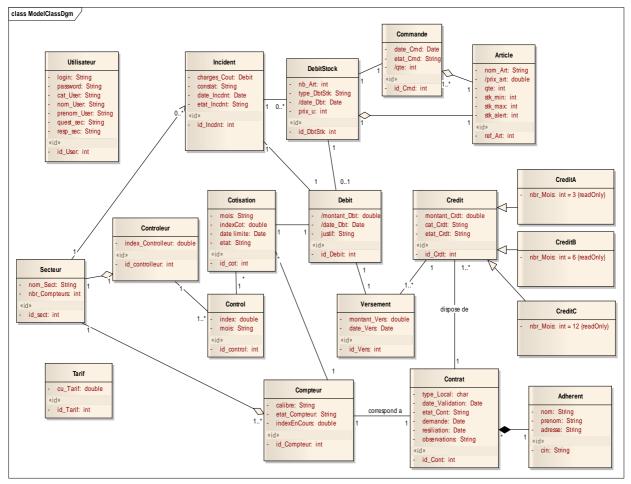


Figure 4: Diagramme de classes

2.2. Modèle dynamique

Le modèle dynamique décrit les interactions, entre les objets du système, lors de la réalisation des cas d'utilisation à l'aide des diagrammes de séquences et d'états. Dans la suite nous allons présenter le modèle dynamique du cas d'utilisation « gestion des consommations ».

Digrammes de séquences

Le digramme de séquence de la figure 5 permet de réaliser l'édition des indexes et la génération des cotisations :

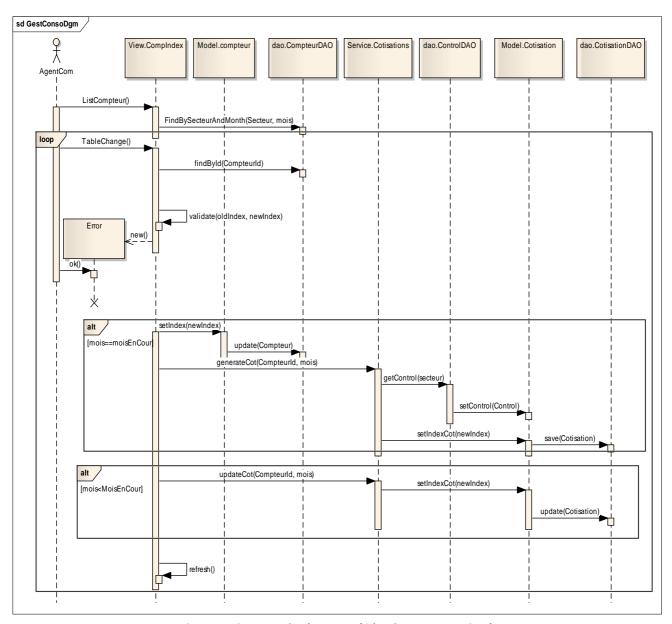


Figure 5 : Diaramme de séquence 1 (Gérer les consommations)

Le digramme de séquence de la figure 6 permet de réaliser le règlement des cotisations :

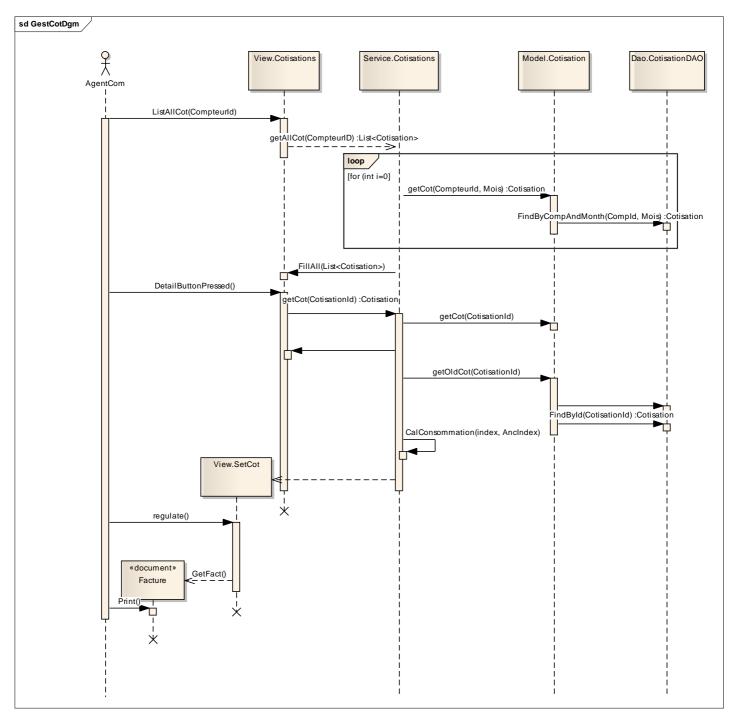


Figure 6 : Diagramme de séquence 2 (Gérer les cotisations)

Digrammes d'états

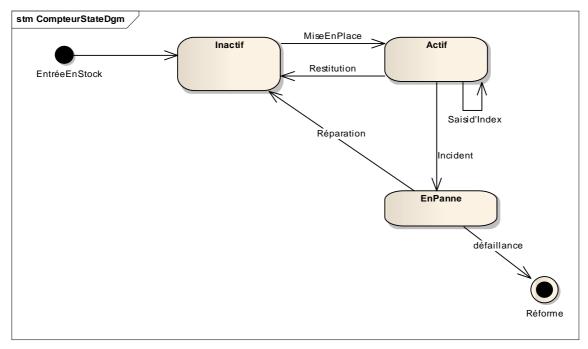


Figure 7: Diagramme d'état 1 (Compteur)

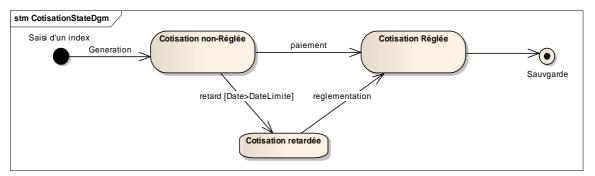


Figure 8: Diagramme d'etat 2 (Cotisation)

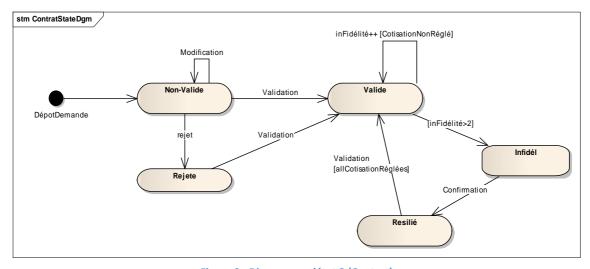


Figure 9 : Diagramme détat 3 (Contrat)

Chapitre 4 : Etude Technique

Dans ce chapitre nous allons présenter l'étude technique de notre projet. L'architecture de notre système sera présentée en premier lieu suivie des outils utilisés lors de la réalisation.

CHAPITRE 4: ETUDE TECHNIQUE

1. Architecture et organisation en couches

Une architecture, dans un développement objet, décrit l'organisation et les interactions des objets du système. Il s'agit de regrouper les objets en classes, les classes en domaines, et les domaines en couches. Construire une architecture consiste à déterminer le nombre de couches et à définir leurs contenus. Dans la suite nous décrivons les deux styles d'architectures les plus utilisées.

- Architecture 2 tiers (Client lourd 1): L'utilisation courante de la majorité des services fournis par l'application se fera essentiellement sur les postes des intervenants se situant dans le local. De ce point s'affirme l'utilité et l'efficacité de cette disposition autant qu'architecture appropriée face à une architecture «Intranet web» peu adaptée à la nature de son utilisation.
- Architecture en 5 couches: Pour avoir une architecture robuste, modulable et évolutive il nous faut utiliser le principe de « couches ». Nous allons donc séparer au maximum les différentes parties de l'application (Données, DAO, Application, Métier, Présentation). Ceci correspond à une architecture en 5 couches:

¹ Le terme « client lourd » (en anglais « heavy client »), par opposition au client léger, désigne une application cliente graphique exécutée sur le système d'exploitation de l'utilisateur. Un client lourd possède généralement des capacités de traitement évoluées et peut posséder une interface graphique sophistiquée. Néanmoins, ceci demande un effort de développement et tend à mêler la logique de présentation (l'interface graphique) avec la logique applicative (les traitements).

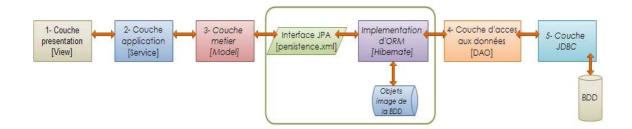


Figure 10: Architecture en couches

2. Conception Générique

Dans ce volet on spécifiera les différents outils techniques déployés pour la réalisation de notre système ainsi que les différentes technologies mises au point pour son élaboration.

Le mapping objet/relationnel (ORM):

La notion d'ORM est le fait d'automatiser la persistance des objets java dans les tables d'une base de données relationnelle, ainsi que son contraire, récupérer les données des tables directement dans des objets java.

Pour implémenter ce concept, un Framework doit être composé des quatre éléments suivants :

- Une API pour effectuer les opérations CRUD (Create, Retrieve,
 Update, Delete) de base sur les objets des classes persistantes
- Un langage ou une API pour créer des requêtes qui se réfèrent aux classes et à leurs propriétés.
- Un moyen simple de spécifier les métadonnées pour le mapping
- L'implémentation de concepts tels que l'automatic dirty checking (mise à jour automatique des tables quand les objets java correspondants sont modifiés), le lazy loading (chargement des propriétés des objets uniquement des qu'ils sont accédés) ainsi que d'autres fonctions d'optimisation.

Java Persistence API et Hibernate :

L'API Java Persistence repose sur des entités qui sont de simples POJO2s annotés et sur un gestionnaire de ces entités (EntityManager) qui propose des fonctionnalités pour les manipuler (ajout, modification suppression, recherche). Ce gestionnaire est responsable de la gestion de l'état des entités et de leur persistance dans la base de données.

Parmi de nombreuses solutions répondant à cette attente (EJB, iBatis, JDO, Cayenne, Castor, etc..), nous avons choisi d'utiliser le framework Hibernate car c'est à l'heure actuelle le framework le plus puissant, le plus complet, et surtout le plus utilisé dans le monde professionnel.

Les entités Bean :

Les entités dans les spécifications de l'API Java Persistence permettent d'encapsuler les données d'une occurrence d'une ou plusieurs tables. Ce sont de simples POJO (Plain Old Java Object). Un POJO est une classe Java qui n'implémente aucune interface particulière ni n'hérite d'aucune classe mère spécifique.

Le bean entity n'a pas à implémenter d'interface particulière mais il doit en plus de respecter les règles de tous Java beans :

Être déclaré avec l'annotation @ Entity

Posséder au moins une propriété déclarer comme clé primaire avec l'annotation @ld

Une propriété particulière est la clé primaire qui sert d'identifiant unique dans la base de données mais aussi dans le POJO. Elle peut être de type primitif ou de type objet. La déclaration de cette clé primaire est obligatoire.

La description du mapping entre le bean entité et la table peut être fait de deux façons :

² (Plain Old Java Object)

L'annotation @ Table permet de lier l'entité à une table de la base de données. Par défaut, l'entité est liée à la table de la base de données correspondant au nom de la classe de l'entité. Si ce nom est différent alors l'utilisation de l'annotation @Table est obligatoire. C'est notamment le cas si des conventions de nommage des entités de la base de données sont mises en place.

Passage du modèle objet au modèle relationnel

L'utilisation d'un SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles. Les deux structures ayant des analogies, les équivalences exprimées au tableau 11-1 sont utilisées pour en réaliser le rapprochement.

Une classe définit une structure de données à laquelle souscrivent des instances; elle correspond donc à une table du modèle relationnel : chaque attribut donne lieu à une colonne, chaque instance stocke ses données dans une ligne (T-uplet) et son OID sert de clé primaire.

Certains attributs de type complexe ne correspondent à aucun des types de SQL; on rencontre fréquemment ce cas pour les attributs représentant une structure de données. Un type complexe peut être conçu;

- soit avec plusieurs colonnes, chacune correspondant à un champ de la structure :
- soit avec une table spécifique dotée d'une clé étrangère pour relier les instances aux valeurs de leur attribut complexe.

Modèle objet	Modèle relationnel
Classe	Table
Attribut de type	simple Colonne
Attribut de type composé	Colonnes ou clé étrangère
Instance	T-uplet
OID	Clé primaire
Association	Clé étrangère ou Table de liens
Héritage	Clé primaire identique sur
	plusieurs tables

Tableau 10 : Transition entre model Objet et model Physique de données

Ainsi et à l'aide d'un outil de passage à savoir Entreprise Architecte on a obtenu à partir du model statique qu'on réaliser auparavant un model physique de données suivant (Figure 21):

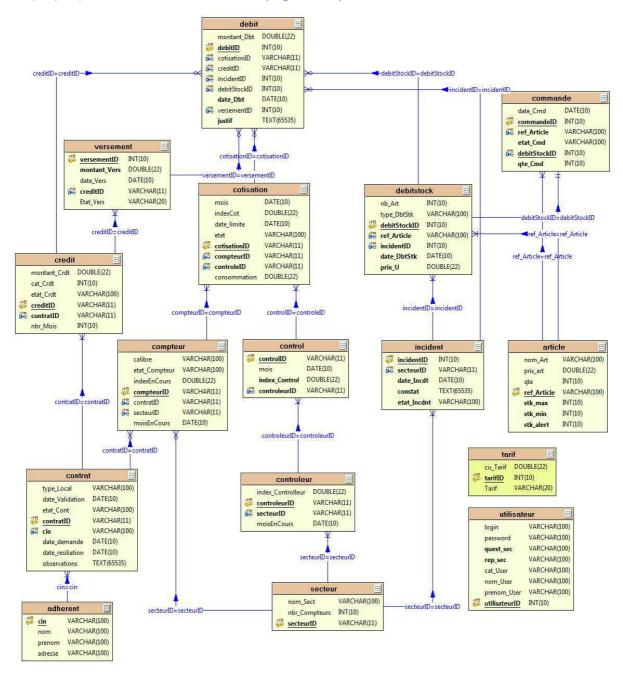


Figure 11 : Model physique de données

3. Outils de réalisation

Pour répondre aux premières contraintes recueillis à partir des besoins techniques primaires exprimés auprès du client on a opté pour un certain nombre d'outils et de technologies qu'on juger les plus adapté à la nature du projet dont on cite :

MySQL server: En un premier lieu on a songé en un SGBD capable de gérer le nombre large et évolutif des enregistrements sans oublier de prendre en considération que l'association est d'un but non lucratif chose qui a nécessité au préalable l'utilisation d'un outil non payant à titre d'exemple ''POSTGREE'', mais pour les essai lors du développement on s'était suffi d'un outil expérimental ''MySQL server''.

Java: Java est un langage autant puissant que facile à utiliser ce qui fait de lui un des solutions les plus utilisé dans le monde de programmation de plus et vu son interopérabilité ce langage s'est imposé comme meilleur solution adapté aux choix que nous avons fixé, ses Interfaces SWT sont très conviviales et faciles a élaboré ce qui nous laissera largement de temps pour le développement des composants du système et les API disponibles pour ce langage nous ont fournis un vaste choix de manipulations.

Java SWT: SWT est un kit graphique propose une approche intermédiaire entre ces deux prédécesseurs (AWT et Swing): utiliser autant que possible les composants du système et implémenter les autres composants en Java. SWT est écrit en Java et utilise la technologie JNI pour appeler les composants natifs. SWT utilise autant que possible les composants natifs du système lorsque ceux sont présentés, sinon ils sont réécrits en pur Java. Les données de chaque composant sont aussi stockées autant que possible dans le composant natif, limitant ainsi les données stockées dans les objets Java correspondant.

Eclipse: Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire, Les principaux modules fournis en standard avec Eclipse concernent Java mais aussi pour d'autres aspects du développement (base de données, conception avec UML, ...).

JDBC: JDBC est l'acronyme de Java DataBase Connectivity et désigne une API définie par Sun pour permettre un accès aux bases de données avec Java.

Hibernate: Hibernate est un projet open source visant à proposer un outil de mapping entre les objets et des données stockées dans une base de données relationnelle. Ce projet ne repose sur aucun standard mais il est très populaire notamment à cause de ses bonnes performances et de son ouverture avec de nombreuses bases de données.

JasperReports: JasperReports est outil 100% Open Source de génération d'états/ rapports développé en Java, pour des applications Java. Il se présente sous la forme de librairies à intégrer aux applications. Ces librairies permettent la visualisation ou l'export de données vers de multiples formats.

JapserReport nous a permis essentiellement de :

Produire des documents types: bons de commande, factures, rapports d'activités, ... bref des documents le plus souvent dynamiques, à partir de données existantes, et ce dans divers formats

Permettre l'export de données à des formats divers pour une application existante (sortie imprimable, mailing ...)

CHAPITRE 5: REALISATION

CONCLUSION