

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Saad Dahleb de Blida
Faculté des Sciences
Département d'Informatique



MÉMOIRE DE FIN D'ÉTUDES POUR L'OBTENTION DU DIPLÔME DE
MASTER EN INFORMATIQUE (LMD)
Option : INGENIERIE DU LOGICIEL

**CONCEPTION ET IMPLEMENTATION D'UN OUTIL
D'EDITION ET DE CALIBRATION DES ETATS DE SORTIE**

Présenté et soutenu publiquement
Le 28.09.11

par Ghiles MOUHEB
et

Abdelmalik Nedjmallah ZOUBIR

Promoteur : M^r BENNOUAR
Copromoteur : M^{lle} FARHI. F.

Président : M^r OULD AÏSSA
Jury : M^{me} BENSETTITI

2010- 2011

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb de Blida

Faculté des Sciences

Département d'Informatique



MÉMOIRE DE FIN D'ÉTUDES POUR L'OBTENTION DU DIPLÔME DE
MASTER EN INFORMATIQUE (LMD)
Option : INGENIERIE DU LOGICIEL

CONCEPTION ET IMPLEMENTATION D'UN OUTIL
D'EDITION ET DE CALIBRATION DES ETATS DE SORTIE

Présenté et soutenu publiquement
Le 28.09.11

par Ghiles MOUHEB
et

Abdelmalik Nedjmallah ZOUBIR

Promoteur : M^r. BENNOUAR
Copromoteur : M^{lle} FARHI. F.

Président : M^r. OULD AÏSSA
Jury : M^{me} BENSETTITI

2010- 2011

REMERCIEMENTS

Nous tenons à remercier tout d'abord notre promotrice Mlle. FARHI. F. pour avoir consacré de son temps pour nous encadrer et nous orienter sur ce travail.

Merci aux membres du jury, d'avoir accepté de juger et critiquer ce fructueux travail.

Merci à toi Malik, d'avoir participé à ce projet avec moi, et d'avoir mis du sérieux pour l'accomplir dans les temps.

Merci à mes parents, et ma sœur qui m'ont toujours soutenu, poussé à aller de l'avant et surtout pour leur patience et leur aide précieuse.

Merci à toute ma famille DENDENE et MOUHEB sur lesquelles j'ai toujours pu compter.

MOUHEB Ghiles

Je remercie Samia de nous avoir aidé et épaulé durant ce travail.

Je remercie mes parents ainsi que mon frère et ma sœur pour leurs soutiens moral continu.

ZOUBIR Abdelmalik Nedjmallah

DEDICACES

Je dédie ce travail.

Aux deux personnes que j'ai tant aimé qu'elles assistent à la fin de mon parcours universitaire, les regrettés Grand père Mohand DENDENE et Grand mère Yamina MOUHEB.

A ma petite famille : mon père Ali, ma mère Nadia, ainsi qu'à ma sœur Samia qui m'a prêté main forte durant ce travail.

A toute ma famille, grands et petits,

A tous mes amis et camarades de classe,

Qu'ils trouvent à travers ce travail ma sincère reconnaissance.

MOUHEB Ghiles

Je dédie ce travail.

A mon père Redouane et ma mère Amina, qui m'ont soutenu tout au long de mon cursus scolaire et qui m'ont procuré le confort nécessaire à l'accomplissement de ce travail.

A mon frère Chouaib et à ma sœur Chahrazed, qui ont su me procurer le calme nécessaire ainsi que me décharger de certaine tâche de la vie courante.

A ma très chère et très regrettée cousine et sœur KADEM fairouz Imen que Dieu l'accueille dans son vaste paradis, qui a toujours voulu me voir réussir et qui m'encourageait dans chaque pas que je faisais dans la vie. Ses conseils m'ont été précieux.

A mes chers amis Sarah et Boualem.

A MOUHEB Ghiles qui a partagé ce travail avec moi et nous nous sommes encouragés mutuellement afin de le finir et le présenter dans les délais.

ZOUBIR Abdelmalik Nedjmallah

Résumé

Notre travail consiste à réaliser un programme qui fait partie du projet eGouvernement.

Motivés par le besoin de l'Assemblée Populaire Communale (APC), pour l'informatisation de ses services, nous avons conçu un programme qui a pour but la réparation des décalages d'impression sur les documents administratifs préimprimés. Pour cela, nous avons développé une application « **Editeur de rapports** », munie d'une interface graphique, qui donnera à l'utilisateur l'accès au rapport et ainsi modifier son contenu. Cet éditeur, permettra non seulement l'arrangement des décalages, mais aussi la possibilité de changer la police, ainsi que de modifier l'emplacement des textes statiques...etc.

Ainsi nous touchons à notre but, qui est de réduire les erreurs dues aux différents formats de feuilles, d'autant plus que l'APC utilise un format de feuille spécifique.

Abstract

Our job is to make a program which is a plan of the eGouvernement.

Motivated by the need for the Communal People's Assembly, for the computerization of its services, we designed a program that aims to repair mismatches printing on pre-printed paperwork. For this, we developed an application « **Editeur de rapports** », equipped with a graphical interface, which will give to user, access to the report and thus modify its contents. This editor allows not only the arrangement of shifts but, also the ability to change the font, as well as change the location of static text ... etc.

Thus we reach our goal of reducing errors due to different sizes of leaves, especially as the APC uses a specific sheet size.

ملخص

مهمتنا هي تحقيق برنامجا، الذي يعتبر جزء من مشروع eGovernment .
بدافع حاجة المجلس الشعبي البلدي لحوسبة الخدمات التي يقدمها، قمنا بتصميم برنامجا يهدف إلى
إصلاح عدم تطابق الطباعة على الأوراق المطبوعة مسبقا. لذا، قمنا بتطوير تطبيقا «Editeur de rapports»
مجهزا بواجهة رسومية، التي تمكن للمستخدم الوصول إلى التقرير وبالتالي تعديل محتوياته
هذا المحزر، لا يسمح فقط بحل إشكالية عدم التطابق، ولكن يمنح أيضا القدرة على تغيير الخط،
وموقع النصوص الثابتة... الخ
وهكذا نصل إلى هدفنا المتمثل في التقليل من الأخطاء بسبب الأوراق ذات الأحجام المختلفة، خاصة
أن المجلس الشعبي البلدي يستخدم أوراقا ذات حجم معين

LISTES DES FIGURES

Figure. 1. Schéma illustrant la Création d'un rapport	20
Figure. 2. Les étapes d'élaboration d'un rapport	21
Figure. 3. Représentation des parties constituante d'un rapport	29
Figure. 4. Cas d'utilisation global	32
Figure. 5. Diagramme de cas d'utilisation gestion de calibrage	33
Figure. 6. Diagramme de classes	38
Figure. 7. Diagramme d'activité globale	39
Figure. 8. Diagramme d'activité gestion de calibrage	40
Figure. 9. Diagramme de séquence	42
Figure.10. Interface d'accueil	46
Figure.11. Edition de texte	47
Figure.12. Recherche du texte.....	48
Figure.13. Les différents thèmes de l'application	49
Figure.14. Propriétés	50
Figure.15. Les différents éléments de propriétés	50
Figure.16. Génération de rapport	51
Figure.17. Aperçu du rapport final	52
Figure.18. La calibration	53
Figure.19. Aperçu du rapport final après rectifications	54
Figure.20. Exportation du rapport au format PDF	55

LISTE DES ABREVIATIONS

APC	: Assemblée Populaire Communale
API	: Applications Programming Interface
CDTA	: Centre de Developpement des Technologies Avancées
CSV	: Comma Separated Value
DTD	: Document Type Definition
HTML	: HyperText Markup Language
JAR	: JavaARchive
JRXML	: JasperReport eXtensible Markup Language
ODF	: Open Document Format
PDF	: Portable Document Format
RTF	: Rich Text Format
SVG	: Scalable Vector Graphics
W3C	: World Wide Web Consortium
XHTML	: eXtensible HyperText Markup Language
XLS	: eXceL Spreadsheet
XML	: eXtensible Markup Langue
XSL-FO	: eXtensible Stylesheet Language – Formatting Objects

Table des matières

REMERCIEMENTS	4
DEDICACES	5
LISTE DES FIGURES	8
LISTE DES ABREVIATIONS	9
TABLE DES MATIERES	10
INTRODUCTION	14
CHAPITRE I. ETAT DE L'ART	16
I.1. Introduction	17
I.2. Reporting	17
<i>I.2.1. Définition d'un état</i>	17
<i>I.2.2. Définition du reporting</i>	17
<i>I.2.3. Principe du reporting</i>	17
<i>I.2.4. Type de reporting</i>	17
<i>I.2.5. Outils du reporting</i>	18
I.2.5.1. Définition	18
I.2.5.2. Quelques outils de reporting	18
<i>I.2.6. Création d'un rapport</i>	19
I. 2.6.1 Java & le reporting	20
I. 2.6.2 Bibliothèques java pour le reporting	21
I.3. XML	22
<i>I.3.1 Définition</i>	22
<i>I.3.2 Syntaxe de XML</i>	22
I.3.2.1 L'entête.....	22
I.3.2.2 Racine	23
I.3.2.3 Eléments et attributs	23
I.3.2.4 Les commentaires	24
I.3.2.5 Représentation d'un document XML valide.....	24
<i>I.3.3 Document XML orienté document et orienté donnée</i>	24
<i>I.3.4 Structure et validation d'un document XML</i>	25

I.4 JRXML	25
<i>I.4.1 Définition</i>	25
<i>I.4.2 Structure d'un fichier JRXML</i>	25
I.4.2.1 Représentation générale d'un fichier JRXML	26
I.4.2.2 Les paramètres	26
I.4.2.3 Les champs	26
I.4.2.4 Les expressions	27
I.4.2.5 Les variables	27
I.4.2.6 Les groupes	28
I.4.2.7 Les requêtes	28
I.4.2.8 Les bandes	28
I.4.2.9 Les différentes parties d'un rapport	29
<i>I.4.3 Manipulation des fichiers JRXML via java</i>	30
I.4.3.1 Chargement et compilation du fichier	30
I.4.3.2 Exécution du rapport	30
I.4.3.3 Création ou exportation du rapport au format PDF	30
I.5 Conclusion	30

CHAPITRE II. CONCEPTION31

II.1 Introduction	32
II.2 Diagrammes de cas d'utilisation	32
<i>II.2.1 Diagramme de cas d'utilisation global</i>	32
<i>II.2.2 Diagramme de cas d'utilisation gestion de calibrage</i>	33
II.3 Diagramme de classes	33
<i>II.3.1 Dictionnaire des données</i>	34
<i>II.3.2 Diagrammes de classes</i>	38
II.4 Diagrammes d'activité	39
<i>II.4.1 Diagramme d'activité global</i>	39
<i>II.4.2 Diagramme d'activité gestion de calibrage</i>	40
II.5 Diagramme de séquence	41
<i>II.5.1 scénario</i>	41
<i>II.5.2 Diagramme de séquence global</i>	42
II.6 Conclusion	43

CHAPITRE III. REALISATION ET IMPLEMENTATION	44
III.1 Introduction	45
III.2 Environnement de Travail	45
III.3 Environnement Logiciel	45
III.4 Méthodologie	45
III.5 Extrait de réalisation	46
<i>III.5.1 Ecran de l'interface d'accueil</i>	46
<i>III.5.2 Edition de texte</i>	47
<i>III.5.3 La recherche du texte</i>	48
<i>III.5.4 Préférences</i>	49
<i>III.5.5 Propriétés de base du rapport</i>	49
III.5.5.1 Les différents éléments de propriétés	50
III.6 Test de l'outil	50
<i>III.6.1 Génération de rapport</i>	51
<i>III.6.2 Aperçu du rapport d'impression</i>	52
<i>III.6.3 Calibration des états</i>	53
<i>III.6.4 Aperçu du rapport d'impression après rectifications</i>	54
<i>III.6.5 Aperçu du rapport d'impression en format PDF</i>	55
CONCLUSION GENERALE	56
GLOSSAIRE	58
REFERENCES BIBLIOGRAPHIQUES	60



INTRODUCTION

INTRODUCTION

On ne peut parler d'informatique, sans parler de l'informatisation d'une activité ou d'une organisation.

L'informatisation est l'un des plus grand pas vers le développement ; assurer ce développement, c'est assurer la prospérité de la société. L'informatisation nous offre une infinité d'avantages, comme par exemple la rapidité du service donné, réduire les coûts, et la chute des prix des ordinateurs facilitera d'avantage l'essor de l'informatisation.

L'Assemblée Populaire Communale ou APC, est l'une des plus grandes infrastructures destinées au service public, son informatisation revient à rendre la vie facile à plus d'un, et pour cela, le Centre de Développement des Technologies Avancées (*CDTA*) a formé des chercheurs, des ingénieurs ainsi que des techniciens, afin d'accomplir ce projet d'un très grand impact sur la société.

L'informatisation au niveau de l'APC se déroule en plusieurs étapes. Notre travail, qui s'inscrit dans le cadre d'eGouvernement , a pour but de mettre en œuvre des programmes informatiques pouvant remédier à l'un des principal problèmes rencontrés au niveau de l'APC et autres structures gouvernementales, situé au niveau des documents administratifs fournis par ces derniers, et ceci en se focalisant plus sur l'une des étapes de cette informatisation, et qui est la création d'un sous système qui prend en charge les documents administratifs pré-imprimés (extraits de naissance, fiches individuelles... etc.).

Lors de l'impression d'un document au niveau des APCs, par exemple des extraits de naissance, La première des choses qui nous vient à remarquer, est le décalage très prononcé des renseignements, ce qui pourrait invalider le document.

Donc notre travail consiste plus exactement à mettre en œuvre un éditeur qui pourra d'une part résoudre ce problème de décalage et d'une autre être utilisé sur des projets autres que le notre.

Cet éditeur se présentera sous forme d'une application qui va permettre aux utilisateurs de manipuler d'une manière fluide les informations présentes dans les différents documents administratifs.

Le but de ce travail est de réaliser un sous système d'impression d'état qui peut facilement s'adapter avec les divers pré-imprimés. Lorsqu'une impression sur papier s'avère non satisfaisante l'opérateur peut déclencher une opération de calibrage qui va lui permettre d'adapter le système au pré-imprimé en cours. Ainsi, ce système de calibrage permettra de faire bouger les champs de textes en différents sens, jusqu'à atteindre un état satisfaisant. Une fois cet état obtenu ; le système l'utilisera par la suite et ne sera remis en cause que si le même problème refait surface.

Ce sous système comprendra un éditeur d'états et une calibration pour chaque état.

Donc les objectifs de ce travail se résument à :

- a) Implémenter une application qui permet la manipulation des formulaires.
- b) Régler les décalages d'une manière simple
- c) Permettre aux utilisateurs une utilisation aisée
- d) Utiliser cette application éventuellement dans d'autres projets.

CHAPITRE I

Etat de l'art

I.1 Introduction

Ce chapitre comportera quelques notions de base utilisées lors de l'apprentissage et le développement de l'application, ainsi que les diverses connaissances acquises lors du stage de formation.

I.2 Reporting

I.2.1 Définition d'un état

Un état est tout simplement un document présentant des informations structurées pour leur visualisation ou leur impression (pas de modification ou de mise à jour des informations possible). (Jaub, 2009)

I.2.2 Définition du reporting

Le terme de "Reporting" désigne une famille d'outils de Business intelligence destinés à assurer la réalisation, la publication et la diffusion de rapports selon un format prédéterminé. Ils sont essentiellement destinés à faciliter la communication de résultats chiffrés ou d'un suivi d'avancement. (Fernandez, 1998-2011)

I.2.3 Principe du reporting

L'outil de reporting assure l'interrogation des bases de données selon les requêtes SQL préparées lors de l'élaboration du modèle. Le rapport peut ensuite être publié sur l'Intranet, périodiquement en automatique ou ponctuellement à la demande. L'outil offre bien entendu des fonctions spécifiques pour l'élaboration du modèle du rapport, des modules de calcul et de présentation (graphiques) afin de concevoir des comptes rendus particulièrement seyants et pertinents. (Fernandez, 1998-2011)

I.2.4 Type de reporting (Fernandez, 1998-2011)

- Suivi d'activité : Le modèle classique de contrôle périodique de l'activité d'une entité placée sous une autorité
- Reporting financier : Suivi des budgets et autres éléments de l'analyse financière, voir la page suivante
- Reporting projet : Suivi d'avancement et communication autour d'un projet relativement conséquent
- Reporting social : Reporting dans le cadre de la Responsabilité Sociale des Entreprises
- Reporting développement durable

1.2.5 Outils de reporting

1.2.5.1 Définition

Les outils de reporting permettent aux analystes et aux décideurs de présenter facilement, pour eux-mêmes, leur direction ou l'extérieur, les données chiffrées de l'entreprise et leur évolution. Le besoin principal consiste à fusionner ces données afin de leur associer des modèles de documents déjà élaborés. (Atol, 2005)

1.2.5.2 Quelques outils de reporting

a) iReport

En ce qui concerne les outils graphiques permettant de concevoir les modèles, le plus performant est sans doute le iReport. C'est un logiciel Open Source entièrement écrit en Java et qui fonctionne à partir de la librairie JasperReports. Il permet de créer tous types de rapports et il inclut la plupart des possibilités offertes par JasperReports. On peut aussi visualiser directement le rendu des rapports sans avoir à programmer la moindre ligne de code Java. En outre, son interface est très intuitive et facile à prendre en main. (Atol, 2005)

a.1.) Avantages : (Anonyme, 2005)

- Outil de reporting très complet.
- Les rapports sont dynamiques, ils peuvent être régénérés régulièrement.
- Graphiquement très abouti.
- Propose une sorti des documents au format : HTML, PDF, CVS, Excel.

a.2.) Inconvénients : (Anonyme, 2005)

- Les rapports doivent être préparés par une personne possédant des connaissances en SQL et des notions de JAVA.
- Il n'est pas utilisable à partir de l'intranet, il ne permet donc pas une configuration globale de l'outil (par groupe d'utilisateurs) pour les administrateurs. La configuration est effectuée au cas par cas.
- Pour élaborer des graphiques comparatifs (confrontation de valeur dans un même graphique) il faut maîtriser le système de script de iReport.

b) OpenReport

OpenReports is a powerful, flexible, and easy to use open source web reporting solution that provides browser based, parameter driven, dynamic report generation and flexible report scheduling capabilities.

b.1.) Avantages : (Anonyme, 2005)

- Outil de reporting sous forme d'application web.
- Assez simple à utiliser.
- Permet une certaine personnalisation des états par l'utilisateur finaux.
- Propose une sorti des documents au format : html, pdf, cvs, excel.

b.2.) Inconvénients : (Anonyme, 2005)

- Moins complet que iReport.
- Les graphiques sont simples.
- Il est impossible dans la version actuelle de créer des graphiques comparatifs (confrontation de valeur dans un même graphique).
- L'interface n'ai pas très conviviale. On ne sait pas toujours ce que l'on touche ni ce que l'on fait.
- Les rapports doivent être préparés par une personne ayant des connaissances en SQL.

I.2. 6 Création d'un rapport (Atol, 2005)

La création d'un rapport ce passe généralement en 4 étapes :

- L'obtention d'un fichier modèle XML (à l'aide d'éditeurs graphiques comme iReport ou OpenReports Designer)
- La construction du rapport à partir du modèle
- Le remplissage des différents champs du rapport avec les données en provenance de diverses sources (bases de données, classes Java, ...)
- L'exportation du résultat dans plusieurs formats possibles (PDF, HTML, ...).

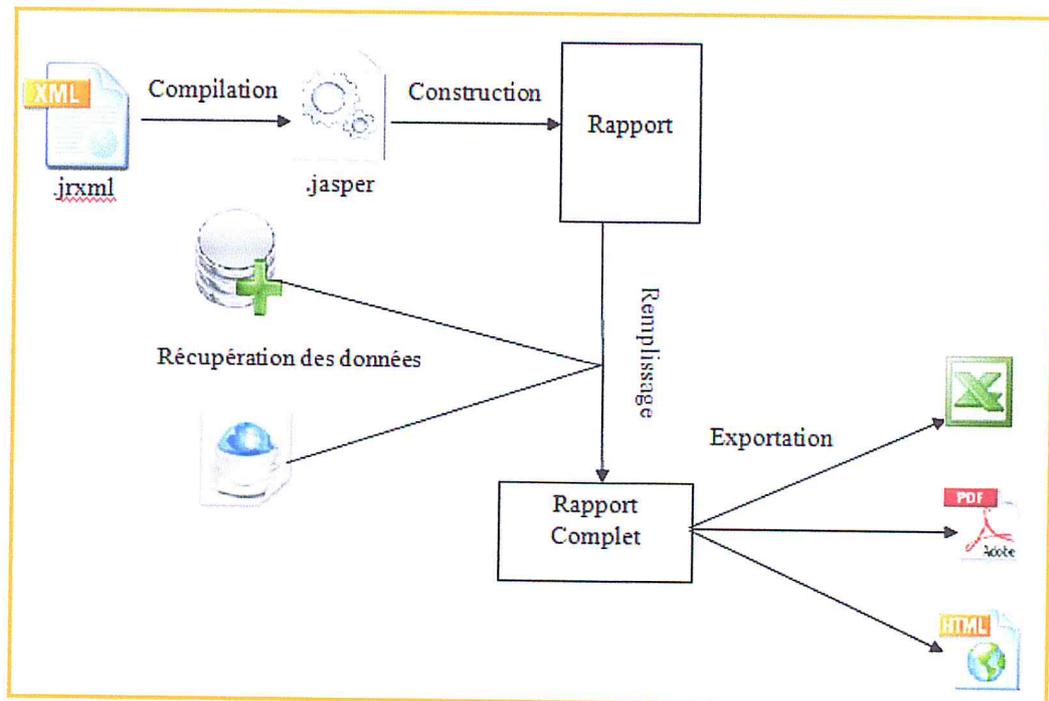


Figure.1. Schéma illustrant la Création d'un rapport

I.2.6.1 Java & le reporting

Avant d'employer ou réaliser un éditeur de rapport, il est nécessaire de détailler les différents objets Java qui entrent en jeu dans le processus de génération d'un rapport, de la conception de celui-ci à la production d'états :

a) JasperDesign

Cet objet représente la définition d'un rapport. Dans la plupart des cas, nous devons créer un objet JasperDesign à partir d'un fichier modèle XML bien qu'il soit également possible de générer ce modèle au moyen de code Java. (Atol , 2005)

b) JasperReport

Cet objet représente un objet JasperDesign compilé. Le processus de compilation vérifie la structure du modèle XML, le compile et le stocke dans un objet JasperReport. (Atol , 2005)

c) JasperPrint

Cet objet représente le rapport final. Un objet JasperPrint est élaboré à partir d'un objet JasperReport par un processus de remplissage qui consiste à insérer dans le rapport des données en provenance d'une source de données quelconque. (Atol , 2005)

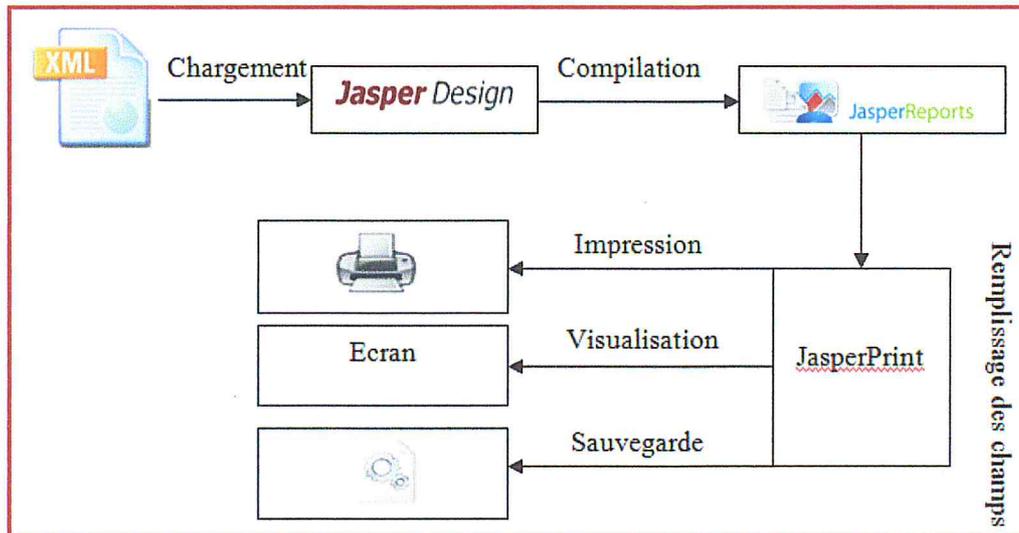


Figure.2. Les étapes d'élaboration d'un rapport

II.2.6.2 Bibliothèques java pour le reporting

Afin de pouvoir employer correctement ces différents objets, nous devons utiliser un certain nombre de bibliothèques (.jar) indispensables au bon fonctionnement du programme Java :

- **Jasperreports.jar** : Cette bibliothèque possède toutes les fonctionnalités propres à JasperReports.
- **commons-digester.jar** : cette bibliothèque permet de transformer un fichier XML en objets Java.
- **commons-collections.jar** :
- **commons-beanutils.jar** : cette bibliothèque apporte une aide à la génération d'objets Java.
- **commons-logging.jar** : cette bibliothèque fournit, de manière transparente, des fonctionnalités de log en utilisant n'importe lequel des frameworks de log existants.

- **Log4j** : C'est une API de journalisation (loggin) destiné à la gestion des journaux pour les applications java. Les journaux offre un historique pertinent du cycle de vie des applications sans aucune intervention de la part de l'utilisateur, tout les messages seront enregistré.
- **JeditSyntax.jar** : Composante zone de texte et ajoute un tas de goodies (sélection rectangulaire, la coloration syntaxique... etc.)
- **Looks_1.1.2.jar** : Permet de donner l'allure (style, thème) à l'application.

I.3 XML

I.3.1 Définition

XML tient pour eXtensible Markup Language : Langage de balisage extensible. C'est un langage qui permet d'échanger des données entres applications hétérogènes, parce qu'il permet de modéliser et de stocker des données de façon portable. XML est un métalangage qui permet de définir de nouvelles balises, ce qui fait de lui un langage extensible.

Il permet encapsuler tout type de données en le représentant sous la forme d'une arborescence. Ce format permet de décrire les données véhiculées mais il ne permet pas de mettre en forme ni d'exploiter celles-ci.

La manipulation et l'exploitation des données inclues dans les fichiers XML se fait de manière fréquente avec les langages XSL et XSLT. De multiples langages de présentation conformes à la syntaxe XML sont en train de voir le jour et de devenir de nouveaux standards (XHTML, XSL-FO, SVG ...) (Anonyme. 2004)

XML et Java ont en commun la portabilité réalisée grâce à une indépendance vis à vis du système et de leur environnement. (Brillant, 2007)

I.3.2 Syntaxe de XML

Un document XML se décompose de plusieurs parties :

I.3.2.1 L'entête

L'entête est représenté dans La première ligne correspond à la déclaration du document. Elle définit la version ainsi que l'encodage (le jeu de caractères) utilisés pour ce document. (Brillant, 2007)

```
<?xml version="1.0" encoding="UTF-8" ?>
```

XML version, comme son nom l'indique représente la version du XML, et encoding représente le jeu de caractères employé. Et cela pour distinguer le rôle de chaque caractère. Certains étant réservés à la syntaxe et d'autres à la représentation des données.

Cette entête est encapsulé par <? et ?> et qu'il n'existe pas d'espaces (de blancs) entre le début du document et cet élément. Autre remarque qui peut surprendre : le prologue n'est pas obligatoire et, dans ce cas, le parseur utilise un comportement par défaut (version 1.0 et encoding UTF-8). (Brillant, 2007)

1.3.2.2 Racine

Les données incluses dans le document sont obligatoirement encadrées par un noeud racine (ROOT) dont le nom n'est pas fixé. Il est donc recommandé d'utiliser un nom décrivant les données contenues. (Anonyme, 2004)

```
<?xml version="1.0" encoding="UTF-8" ?>
<racine>
...
</racine>
```

1.3.2.3 Eléments et attributs

a) Eléments

Chaque nœud du document XML est appelé ELEMENT. Un élément peut lui même contenir d'autres éléments (notion de parent - enfant) et ce sans limite. Cependant :

- Les éléments doivent toujours être fermés (<element></element> ou <element />)
- Ils ne doivent jamais s'entrecroisés (<u><i>syntaxe incorrecte</u></i>) (Anonyme,2004)

b) Attributs

Un attribut est un couple (clé, valeur) associé à la définition d'un élément. Il est localisé dans la balise ouvrante de l'élément. Un élément peut donc avoir de 0 à n attributs uniques. L'attribut est complémentaire de l'élément de par son rôle au sens où il ajoute une information à l'élément ou bien encore le complète dans sa définition.

```
<element attribut1="valeur1" attribut2="valeur2">
    <sousElement attribut3="valeur" >
</element>
```

I.3.2.4 Les Commentaires

```
<!-- Commentaires XML -->
```

I.3.2.5 Représentation d'un document XML valide

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Racine>
  <!-- Commentaire -->

  <LIVRE nom="Livre 1">
    <CHAPITRE numero="1" titre="titre" />
    <CHAPITRE numero="2" titre="titre" />
    <CHAPITRE numero="3" titre="titre" />
  </LIVRE>
  <LIVRE nom="Livre 2">
    <CHAPITRE numero="1" titre="titre" />
    <CHAPITRE numero="2" titre="titre" />
    <CHAPITRE numero="3" titre="titre" />
  </LIVRE>
</Racine>
```

I.3.3 Document XML orienté document et orienté donnée

Lorsque les données sont élaborées par des personnes, on dit que les fichiers XML produits sont orientés document. Lorsque les données sont construites automatiquement par des programmes, on dit que les fichiers XML sont orientés données.

Un fichier XML orienté document peut être, par exemple, un livre, un article, un message.... Par contre un fichier XML orienté donnée peut être par exemple, un sous-ensemble d'une base de données.

Il faut noter que l'élaboration des fichiers XML nécessite des moyens de contrôle et d'édition plus ou moins sophistiqués. On n'utilisera pas pour fabriquer un ouvrage en XML un éditeur trop rudimentaire (comme le bloc-notes sous l'environnement Windows). (Brillant, 2007)

I.3.4 Structure et validation d'un document XML

On associe à un document XML un schéma, qui peut être vu comme le schéma d'une base de données relationnelle. La validation d'un document XML garantit que la structure de données utilisée respecte ce schéma. On peut faire l'analogie avec le respect des règles d'orthographe et de grammaire d'une langue. Les documents XML qui circulent doivent ainsi être en accord avec ce schéma pour être acceptés par la plate-forme. Dans le cas contraire ils sont rejetés et doivent être refaits. Lorsque les flux d'échanges sont denses, la validation peut présenter pour inconvénient de consommer des ressources. Il est difficile de raisonner pour tous les cas, mais la validation peut être considérée comme incontournable à certaines étapes de préparation du cadre d'exploitation. Lorsque les flux sont considérés comme stables, il est alors possible de pratiquer une forme d'assouplissement des règles dans l'optique d'améliorer les performances. (**Brillant, 2007**)

I.4 JRXML

I.4.1 Définition

JRXML est le nom que nous utilisons pour faire référence à des fichiers XML qui représentent la définition d'un modèle de JasperReports (JR) et qui sont conformes à la structure mentionnée DTD. (**Danciu et Chirita, 2007**)

I.4.2 Structure d'un fichier JRxml

Un fichier JRxml est défini par des séries de sections, certaines, concernent les caractéristiques physiques du report, par exemple la dimension de la page, le positionnement des champs, et la hauteur des bandes ; et d'autres concernent les caractéristiques logiques, exemple : la déclaration des paramètres et variables, et la définition des requêtes pour la sélection des données (**Toffoli, 2007**). La structure de ce fichier est déclarée dans un fichier au format DTD (Document Type Definition).

Document Type Definition est un ensemble de déclarations de balisage qui définissent un type de document pour la famille de langages de balisage SGML. Un DTD est une sorte de schéma XML. Les DTDs utilisent une syntaxe concise formelle qui déclare précisément quels éléments et références peuvent apparaître et où dans le document de type particulier, et ce que le contenu des éléments et attributs sont. (**Miller et al., 2010**)

I.4.2.1 Représentation générale d'un fichier JRXml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Created with iReport - A designer for JasperReports -->
<!DOCTYPE jasperReport PUBLIC "-//JasperReports//DTD Report
Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">
<jasperReport name="nomDuRapport" ....>
<jasperreports>
```

I.4.2.2 Les paramètres

Les paramètres sont des références à des objets qui sont passés au rapport au moyen de méthodes de remplissage écrites en Java. Déclarer un paramètre est très simple et requiert uniquement un nom et un type :

```
<parameter name="Titre" class="java.lang.String"></parameter>
```

Les paramètres sont très utiles afin de transmettre des données qui n'apparaissent pas dans les sources de données. Par exemple, nous pouvons passer au rapport le nom de l'utilisateur qui l'a généré ou bien modifier dynamiquement le titre du rapport. (Atol , 2005)

I.4.2.3 Les champs

Les champs représentent la seule façon de récupérer les données à partir d'une source de données et de les transférer ensuite aux différentes routines gérant le rapport. Ces champs doivent avoir obligatoirement le même nom et le même type que les objets correspondants qui résultent de la requête réalisée sur la source de données. (Atol , 2005)

```
<field name="Numero" class="java.lang.Integer"></field>
<field name="Nom" class="java.lang.String"></field>
<field name="Prenom" class="java.lang.String"></field>
<field name="Salaire" class="java.lang.Double"></field>
<field name="DateEmbauche" class="java.util.Date"></field>
```

I.4.2.4 Les expressions

Les expressions sont un dispositif très important de la librairie JasperReports. En effet, elles peuvent être employées pour déclarer des variables qui exécuteront divers calculs, pour spécifier le contenu des champs du rapport ou pour personnaliser l'apparence des différents objets constituant le rapport. Fondamentalement, toutes les expressions sont des expressions Java qui référencent des champs ou des variables du rapport. (Atol , 2005)

```
<variableExpression>, <initialValueExpression>, <groupExpression>,  
<printWhenExpression>, <imageExpression>, <textFieldExpression>
```

Afin d'utiliser des champs dans une expression, le nom du champ doit être placé entre les deux séquences de caractères suivantes : `#{` et `}`.

Par exemple, on décide d'afficher sur le rapport la concaténation du nom et du prénom des employés. (Atol, 2005)

```
<textFieldExpression>  
  #{Nom} + " " + #{Prenom}  
</textFieldExpression>
```

I.4.2.5 Les variables

Les variables permettent de contenir une expression et peuvent être réutilisées.

Elles s'écrivent comme dans certains langages de programmation (php, perl) avec le signe `$`. (Lecomte et Boulanger, 2008)

Elles peuvent être employées pour simplifier la conception du rapport en déclarant seulement une fois une expression qui sera utilisée à plusieurs reprises tout au long du modèle XML. Elles servent également à réaliser certains calculs dans les expressions où elles sont employées. (Atol , 2005)

Exemple : La variable ci-dessous permet de faire la somme totale des salaires

```
<variable name="Somme" class="java.lang.Double" calculation="Sum">  
  
  <variableExpression>#{Salaire}</variableExpression>  
  
</variable>
```

I.4.2.6 Les groupes (Atol, 2005)

Les groupes représentent une façon simple d'organiser les données à l'intérieur d'un rapport. Lorsqu'il remplit un rapport, le moteur de JasperReports teste toutes les définitions de groupes pour s'assurer qu'il n'y a pas de rupture de groupe et pour voir si les sections suivantes sont bien présentes pour chaque groupe :

```
<group name="Employes">
  <groupExpression>...</groupExpression>
  <groupHeader>
  ...
  </groupHeader>
  <groupFooter>
  ...
  </groupFooter>
</group>
```

I.4.2.7 Les requêtes

Lorsque la source de données est une base de données, il est possible d'effectuer une requête directement dans le fichier modèle XML. (Atol, 2005)

```
<queryString>
  SELECT Nom, Prenom FROM Employes
</queryString>
```

I.4.2.8 Les bandes

Chaque section est une partie du rapport bien particulière qui possède une largeur et une hauteur spécifiques et qui peut contenir des objets comme des lignes, des rectangles, des images ou du texte. Pour spécifier le contenu et la disposition d'une section du rapport, il est indispensable d'utiliser l'élément générique band. (Atol, 2005)

```
<band>...</band>
```

Les bandes peuvent inclure de multiples éléments, chacun d'eux étant identifié par une position, une taille et une valeur. (Atol, 2005)

```
<staticText>, <textField>, <line>, <rectangle>, <image>
```

I.4.2.9 Les différentes parties d'un rapport

Le rapport est composé de 9 parties :

- **Background** : Il s'agit de l'arrière plan de la page.
- **Title** : Il s'agit du titre il n'est affiché qu'une fois en haut de la première page.
- **PageHeader** : Cette partie est affichée en haut de toutes les pages du rapport, contrairement à Title.
- **ColumnHeader** : Définit les entêtes des colonnes d'un tableau, elles sont répétées sur chaque page où le tableau est présent.
- **Détail** : Il représente le cœur du rapport, Détail contient les données issues d'une requête SQL.
- **ColumnFooter** : Est affichée une fois en bas de chaque colonne, répétée sur chaque page si nécessaire.
- **PageFooter** : Affichée en bas de toutes les pages du rapport.
- **LastPageFooter** : N'est affichée qu'une fois en bas de la première page du rapport.
- **Summary** :
- **No-Data** : C'est ce qui sera affiché si la requête SQL ne renvoie rien.

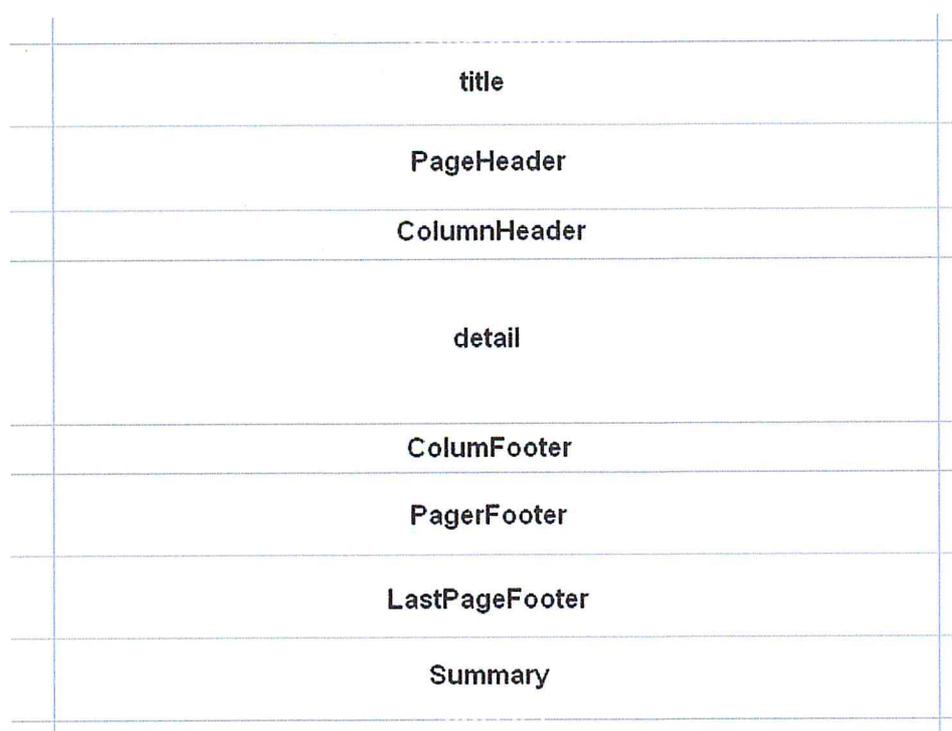


Figure.3. Représentation des parties constituant d'un rapport

I.4.3 Manipulation des Document JRXml via Java

I.4.3.1 Chargement et compilation du fichier

```
JasperDesign jd = JRXmlLoader.load("D:\\chemin_fichier\\Fichier.jrxml");  
JasperReport jr= jasperCompileManager.CompileReport(jd);
```

La première instruction charge le fichier JRXml et le stocke dans un objet de type JasperDesing.

La deuxième instruction, compile l'objet jd (JasperDesign) pour vérifier si la syntaxe du fichier JRXml est respectée.

I.4.3.2 Exécution du rapport

```
JasperPrint jp = JasperFillManager.fillReport(jr,parameters,connetion);
```

Cette instruction permet d'exécuter le rapport en le remplissant avec des données grâce aux paramètres de connexion à la base de données.

I.4.3.3 Création ou exportation du rapport au format PDF

```
JasperExportManager.exportReportToPdfFile(jp, "D:\\Chemin_exportation\\  
Fichier.pdf);
```

A l'aide de la production des instances de la classe précédente les données peuvent être transférées sur le réseau, stockées sous une forme sérialisée sur le disque ou exportées vers différents autres formats tels que HTML, CSV, XLS... etc.

I.5 Conclusion

Ces notions permettent une compréhension globale de la problématique et de l'objectif, ainsi que les différentes démarches essentielles suivies pour accomplir notre travail.

CHAPITRE II

Conception

II.1 Introduction

Modéliser un système avant sa réalisation permet de mieux comprendre le fonctionnement du système. C'est également un bon moyen de maîtriser la complexité d'un système et d'assurer sa cohérence.

Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. *UML* définit des diagrammes structurels et comportementaux pour représenter respectivement des vues statique et dynamique d'un système. (Booch et al., 2003)

Nous avons utilisé 4 diagrammes parmi les neufs proposés par *UML*

- Les cas d'utilisation pour définir les besoins de notre système.
- Le diagramme de classes, pour exprimer de manière générale la structure du système.
- Le diagramme de séquence, pour représenter les interactions entre les objets
- Le diagramme d'activité pour représenter le comportement d'une opération en termes d'actions.

II.2 Diagrammes de cas d'utilisation

II.2.1 Diagramme de cas d'utilisation global

Dans ce cas le diagramme utilise deux types d'utilisateurs :

- 1- OEC principal (Officier d'état civil principal)
- 2- Le citoyen

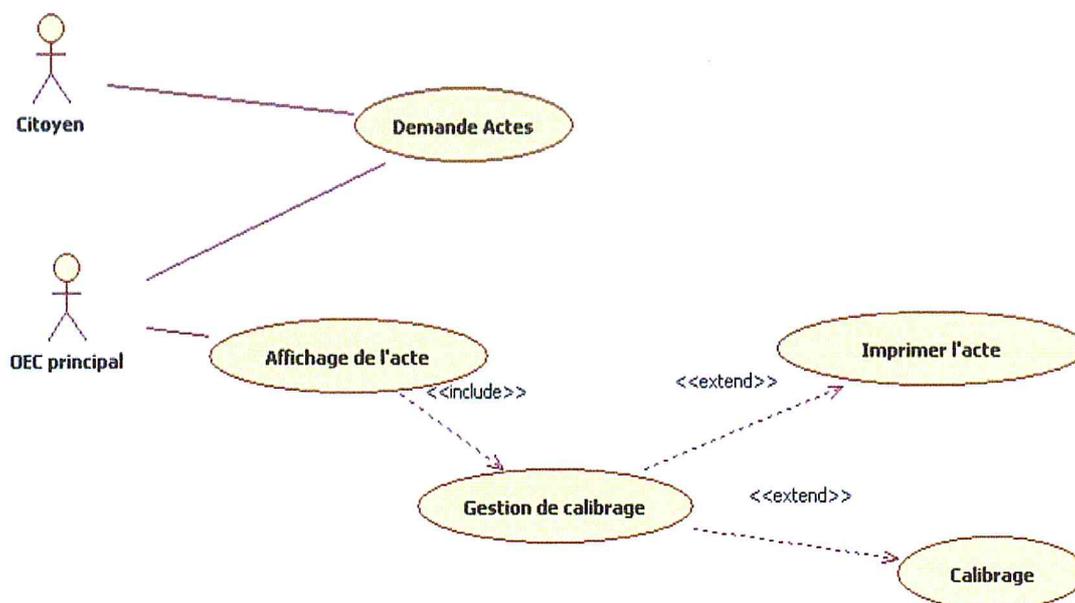


Figure.4. Cas d'utilisation global

II.2.2 Diagramme de cas d'utilisation gestion de calibrage

Dans ce diagramme nous allons représenter la partie de gestion des calibrages.

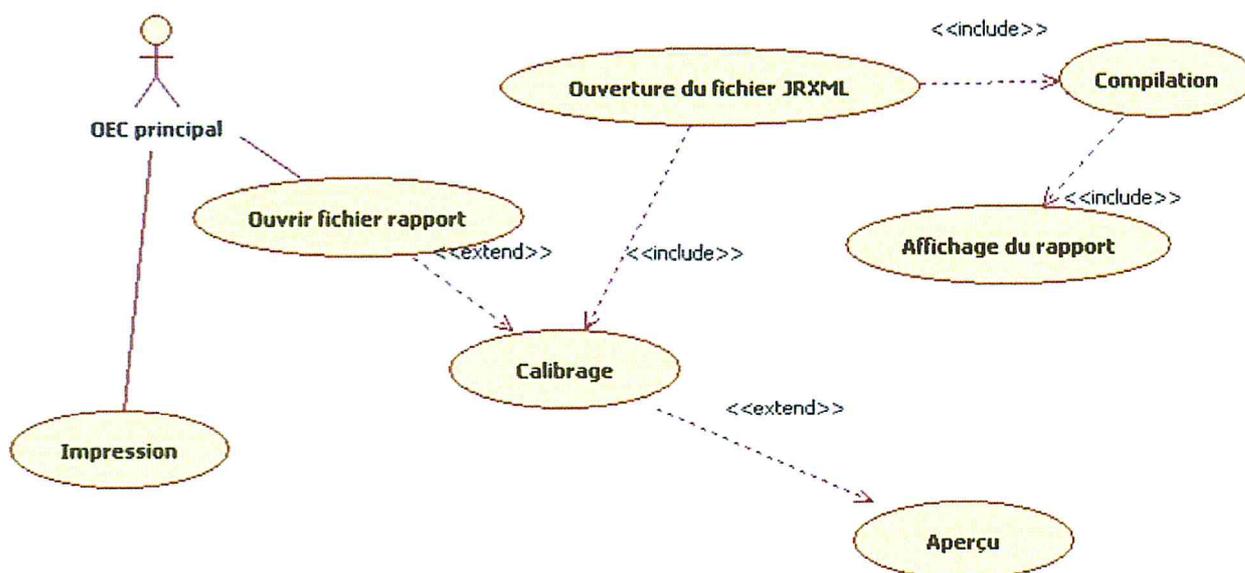


Figure.5. Diagramme de cas d'utilisation gestion de calibrage

L'OEC principal a la possibilité de gérer les calibrages comme suit :

- Ouvrir un fichier de rapport.
- Calibrer le rapport grâce à l'outil de calibration.
- Afficher un aperçu.
- Procéder à une impression du rapport

II.3 Diagramme de classes

Les diagrammes de classes expriment de manière générale la structure statique d'un système (Muller et Gaertner, 2005). Il met en évidence les classes et les relations entre elles : dépendance, association, généralisation, etc. (Laugié, 2008)

II.3.1 Dictionnaire des données

Classe	Attributs	Code : type	Valeur	Les méthodes
Personne	Identifiant de la personne Nom en arabe Prénom en arabe Sexe Profession Adresse Nom en français Prénom en français	#id_p : Chaîne +nom_p_Ar : Chaîne +prenom_p_Ar : Chaîne +sexe_p : Chaîne +prof_p : Chaîne +address_p : Chaîne +nom_p_Fr : Chaîne +prenom_p_Fr : Chaîne	Non nul Non nul Non nul M / F / / Non nul Non nul	Ajouter_p() Modifier_p() Supprimer_p() Rechercher_p() Consulter_p()
Utilisateur	Nom d'utilisateur Mot de passe E_mail Etat de compte	#nom_u : Chaîne +mot_pass_u : Chaîne +email_u : Chaîne +etat_u : Chaîne	Non null Non null / Activé/désactivé	Ajouter_u() Modifier_u() Supprimer_u() Rechercher_u() Consulter_u() Activer_u() Désactiver()
Profil	Identifiant de profil Nom de profil	#id_pr : Entier +non_pr : Chaîne	Non nul Non nul	Ajouter_pr() Modifier_pr() Supprimer_pr() Rechercher_pr() Consulter_pr()
Action	Nom d' action	#nom_ac : Chaîne	Non nul	Rechercher_ac() Consulter_ac()
SouAction	Nom de sous action	#nom_sou_ac : Chaîne	Non nul	/

Commune	Identifiant de commune Nom de commune	#id_com :Chaine +nom_com :Chaine	Non nul Non nul	/
Daira	Identifiant de daïra Nom de daïra	#id_daira :Chaine +nom_daira :Chaine	Non nul Non nul	/
Wilaya	Identifiant de wilaya Nom de wilaya	#id_wilaya : Chaine +nom_wilaya :Chaine	Non nul Non nul	/
Naissance	Identifiant de naissance Date de naissance Heure de naissance	# id_naiss : Chaine +date_nais : Chaine +Heure_naiss :Chaine	Non nul Non nul Non nul	/
Déclaration_ naissance	Identifiant déclaration Numéro de secret Date de déclaration	#id_dec_naiss :Chaine +n°_sec_naiss :Chaine +date_dec_naiss :Date	Non nul Non nul Non nul	Ajouter_dec() Modifie_dec() Supprimer_dec() Recherche_dec() Consulter_dec() Valider_dec()
Demande_ extrai_ naissance	Identifiant demande Numéro de secret Date de demande	#id_dem_naiss :Chaine +n°_sec_naiss :Chaine +date_dem_naiss :Date	Non nul Non nul Non nul	Ajouter_dem() Modifie_dem() Supprimer_dem() Recherche_dem() Consulter_dem() Valider_dem()

Déclaration_mariage	Identifiant déclaration Numéro de secret Date de déclaration	#id_dec_mar :Chaine +n°_sec_mar :Chaine +date_dec_mar :Date	Non nul Non nul Non nul	Ajouter_dec() Modifie_dec() Supprimer_dec() Recherche_dec() Consulter_dec() Valider_dec()
Demande_acte_mariage	Identifiant demande Numéro de secret Date de demande	#id_dem_mar :Chaine +n°_sec_mar :Chaine +date_dem_mar :Date	Non nul Non nul Non nul	Ajouter_dem() Modifie_dem() Supprimer_dem() Recherche_dem() Consulter_dem() Valider_dem()
mariage	Identifiant de mariage Date de mariage Heure de mariage	# id_mar : Chaine +date_mar : Chaine +Heure_mar :Chaine	Non nul Non nul Non nul	/
Déclaration_divorce	Identifiant déclaration Numéro de secret Date de déclaration	#id_dec_div :Chaine +n°_sec_div :Chaine +date_dec_div :Date	Non nul Non nul Non nul	Ajouter_dec() Modifie_dec() Supprimer_dec() Recherche_dec() Consulter_dec() Valider_dec()
Demande_acte_divorce	Identifiant demande Numéro de secret Date de demande	#id_dem_div :Chaine +n°_sec_div :Chaine +date_dem_div :Date	Non nul Non nul Non nul	Ajouter_dem() Modifie_dem() Supprimer_dem() Recherche_dem() Consulter_dem() Valider_dem()

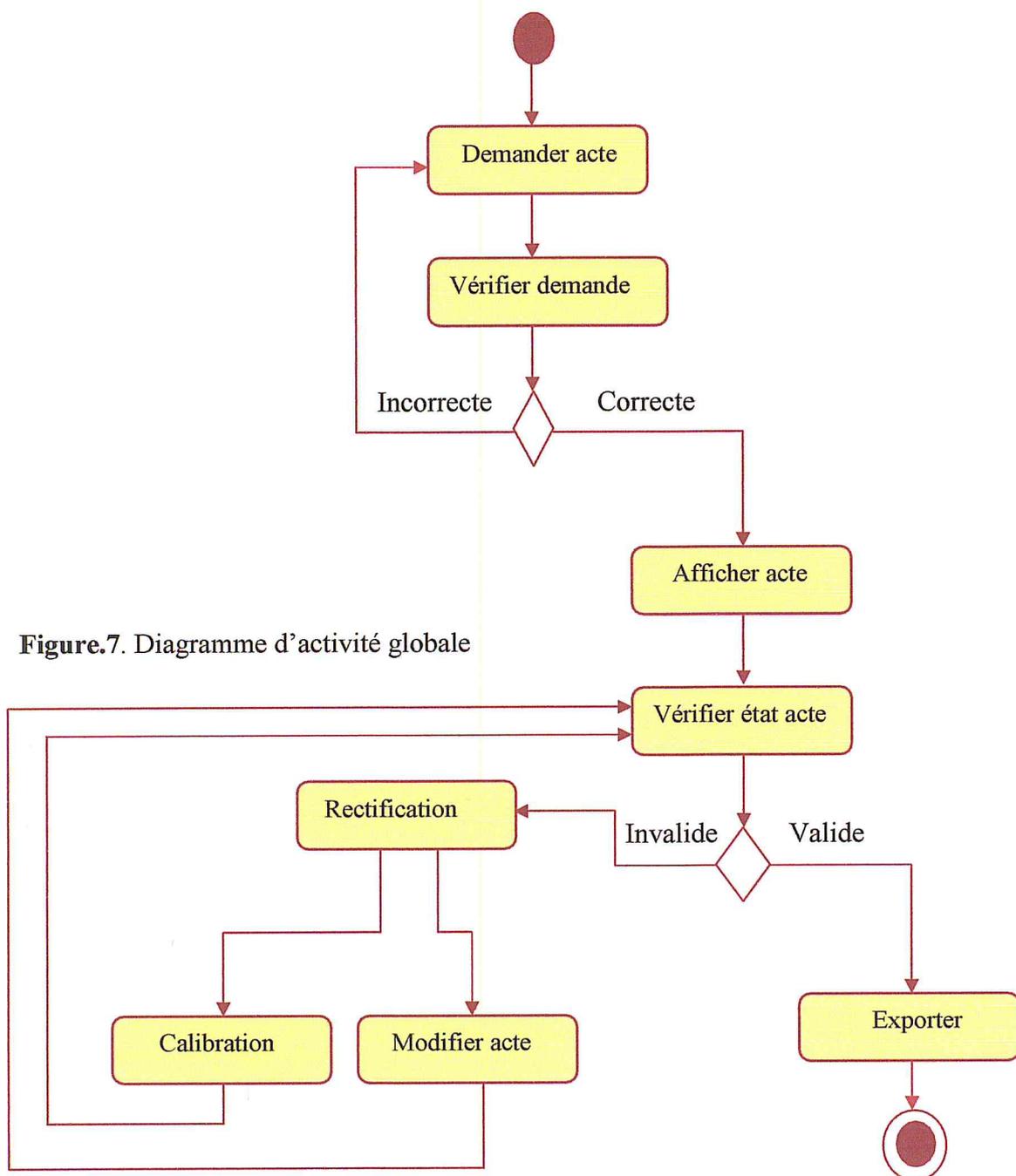
divorce	Identifiant de divorce Numéro de jugement Tribunal de jugement Date de jugement Heure de jugement	# id_div : Chaîne +n_jug : Entier +tribunal_jug :Chaîne +date_jug : Date +Heure_mar :Chaîne	Non nul Non nul Non nul Non nul /	/
Déclaration_décès	Identifiant déclaration Numéro de déclaration Date de déclaration	#id_dec_dec :Chaîne +n°_sec_dec :Chaîne +date_dec_dec :Date	Non nul Non nul Non nul	Ajouter_dec() Modifie_dec() Supprimer_dec() Recherche_dec() Consulter_dec() Valider_dec()
Décès	Identifiant de décès Date de décès Heure de décès	# id_dec : Chaîne +date_dec: Date +Heure_dec :Chaîne	Non nul Non nul Non nul	/

II.4 Diagrammes d'activité

Le diagramme d'activité est basé sur le diagramme d'états-transitions. Il s'agit d'une forme spécifique du diagramme d'états-transition dans lequel chaque état est associé à une activité et toutes les transitions (appelées enchaînements) sont automatiques. (Debrauwer et Van der Heyde, 2008)

Ces diagrammes permettent de représenter la chronologie de différentes actions (activités) intervenant dans une méthode ou un cas d'utilisation. (Bigand, 2006)

II.4.1 Diagramme d'activité globale



II. 4.2 Diagramme d'activité gestion de calibrage

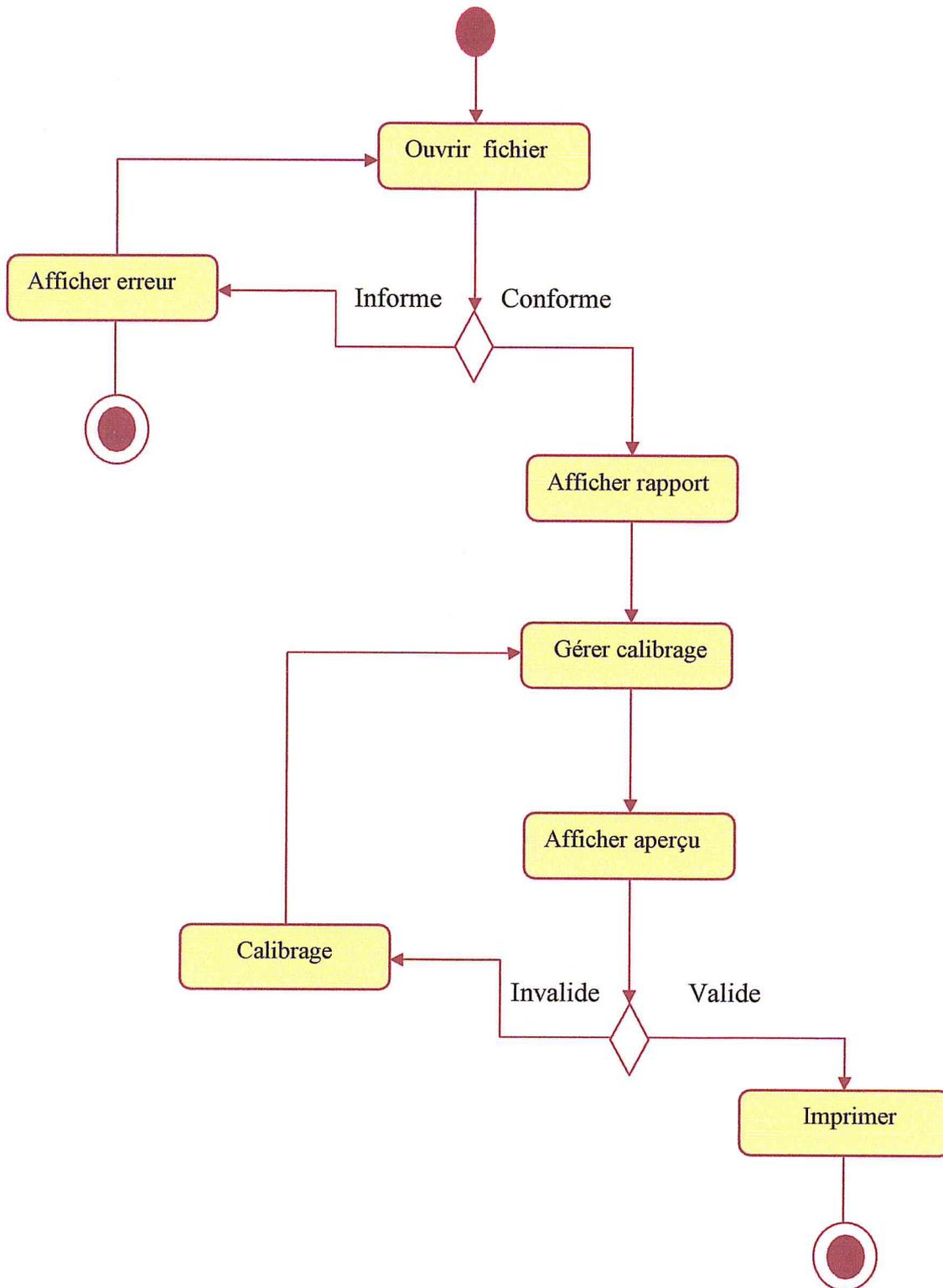


Figure.8. Diagramme d'activité gestion de calibrage

II.5 Diagramme de séquence

Le diagramme de séquence décrit la dynamique du système. A moins de modéliser un très petit système, il est difficile de représenter toute la dynamique d'un système sur un seul diagramme. Aussi la dynamique globale sera représentée par un ensemble de diagrammes de séquence, chacun étant généralement lié à une sous fonction du système. Il décrit aussi les interactions entre un groupe d'objets en montrant de façon séquentielle, les envois de messages qui interviennent entre les objets. Le diagramme peut également montrer les flux de données échangés lors des envois de message. (Debrauwer et Van der Heyde, 2008)

Le diagramme de séquence illustre graphiquement les interactions entre les éléments du système sur un espace en deux dimensions : (Briol, 2008)

- 1- le sens vertical du diagramme précise l'aspect temporel des interactions
- 2- et son sens horizontal précise les interactions entre les éléments impliqués

II.5.1 scénario

- 1) Le Citoyen envoie une demande d'acte de naissance.
- 2) OEC Principal effectue une vérification du rapport.
- 3) OEC principal souhaite faire des calibrages.
- 4) Système affiche le rapport à calibrer.
- 5) OEC principal ouvre le gestionnaire de calibrages.
- 6) OEC effectue les calibrages et modification.
- 7) Le système prend en charge les modifications.
- 8) OEC principal effectue une exportation du rapport s'il n'y a plus de décalage.
- 9) Le citoyen récupère l'acte

II.5.2 Diagramme de séquence

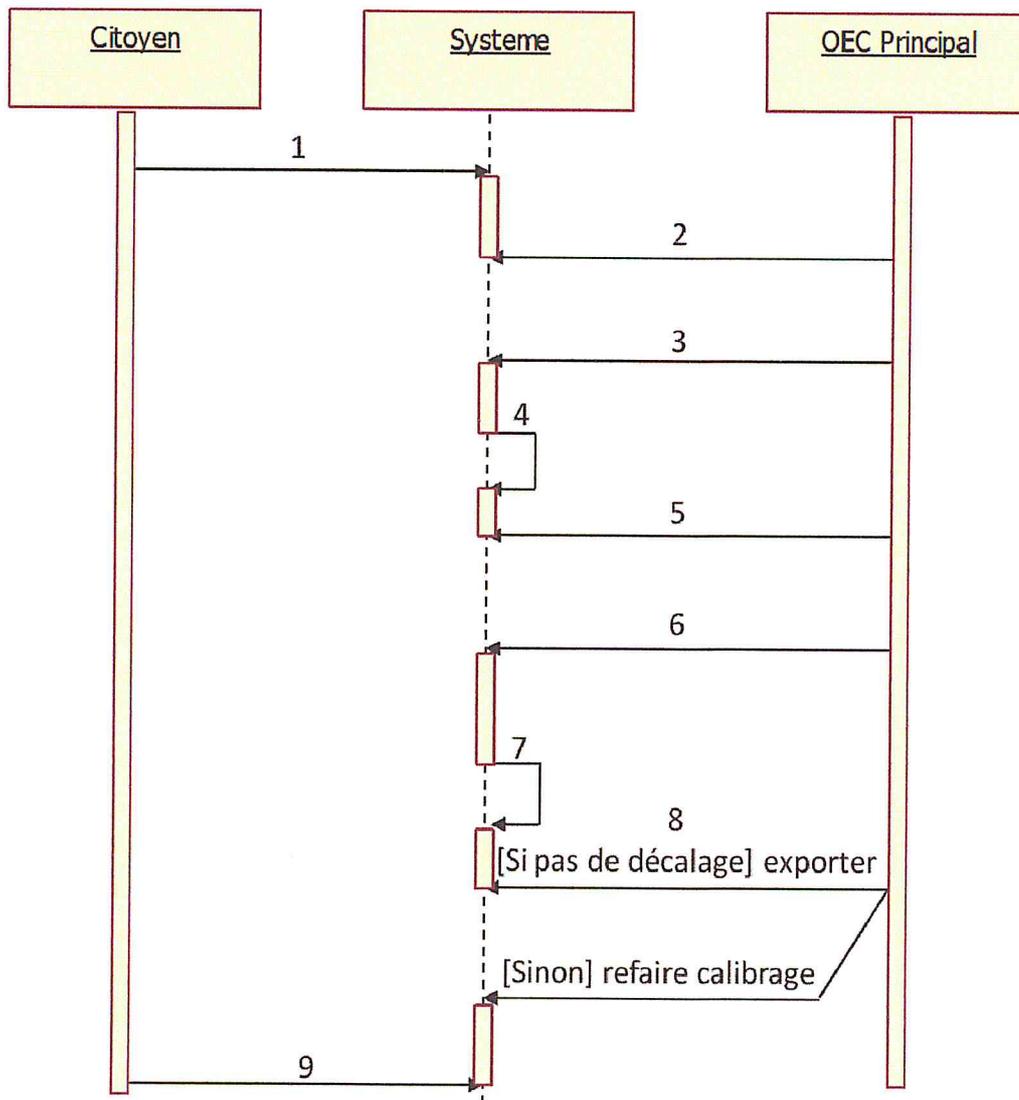


Figure. 9. Diagramme de séquence global

II.6 Conclusion

L'ensemble des diagrammes présentés dans ce chapitre nous ont été très utiles pour mieux comprendre le but de notre travail ainsi que de l'aborder d'une manière correcte.

CHAPITRE III

Réalisation et implémentation

III.1 Introduction

Dans ce chapitre, seront décrits l'environnement matériel ainsi que l'environnement logiciel dans lesquels notre application a été développée. Quelques écrans représentant les fonctionnalités de notre application seront détaillées au fur et à mesure avant d'arriver à l'étape primordiale dans le cycle du logiciel, et qui s'agit de l'étape de test.

Pour tester l'application, la génération d'un Extrait d'acte de naissance est nécessaire afin d'essayer de calibrer et de modifier les états.

III.1 Environnement de Travail

Ce travail a été réalisé au niveau du Centre de Développement des Technologies Avancées (CDTA) : Architecture des systèmes, durant la période comprise entre le mois de Mars et le mois de septembre 2011.

III.3 Environnement Logiciel

Notre projet a été réalisé dans l'environnement logiciel suivant :

- Windows Vista et XP comme Système d'exploitation.
- Eclipse SDK-3.5.2 comme environnement de développement.
- Java version 1.6.0_07 comme machine virtuelle.
- iReport-3.0.0 et iReport-4.0.0 comme générateur de rapports.
- MySQL comme gestionnaire de base de données.

III.4 Méthodologie

La méthodologie suivie dans ce travail, est basée sur la programmation intensive, itérative et progressive. Cette méthodologie incite à commencer par les aspects les plus simples, les réaliser, les tester et passer ensuite à la réalisation d'un autre aspect.

Ce processus pratique permettra d'une part de maîtriser l'outil de programmation et d'avoir à chaque étape une version fonctionnelle d'une partie du logiciel.

La mise en place d'un ensemble de jeux d'essais ou des tests automatiques pour toutes les fonctionnalités développées est importante dans ce travail, afin de garantir au produit un niveau de robustesse élevé.

III.5 Extrait de réalisation

III.5.1 Ecran de l'interface d'accueil

L'interface de notre application est entièrement développée avec l'API (Applications Programming Interface) Swing de java.

Elle permet de fournir toutes les fonctionnalités que notre application offre. (figure.10)



Figure.10. Interface d'accueil

III.5.2 Edition de texte

Notre application offre aussi une interface d'édition de texte, l'utilisateur aura la possibilité de charger un fichier (.jrxml, xml ou jasper) il suffirait juste d'appuyer sur **فتح Ctrl-O**, il peut aussi copier et coller du texte, ainsi que de créer son propre fichier. (figure.11)

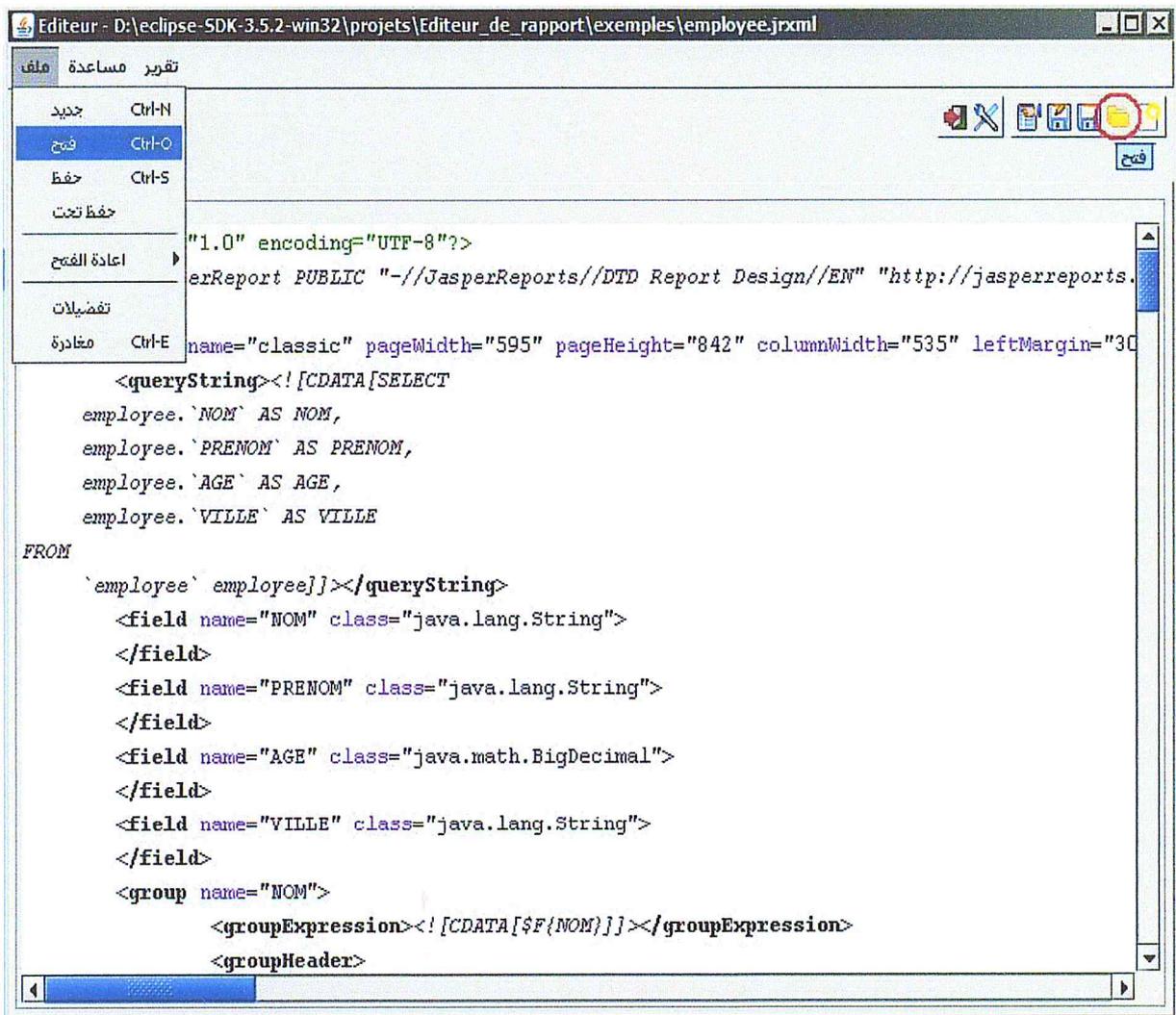


Figure.11. Edition de texte

III.5.3 La recherche du texte (figure.12)

L'utilisateur pourra à tout moment, grâce à la petite fenêtre (Trouver Texte) rechercher un mot, une phrase ou bien un paragraphe pour y accéder en cliquant sur le bouton إيجاد Ctrl-F

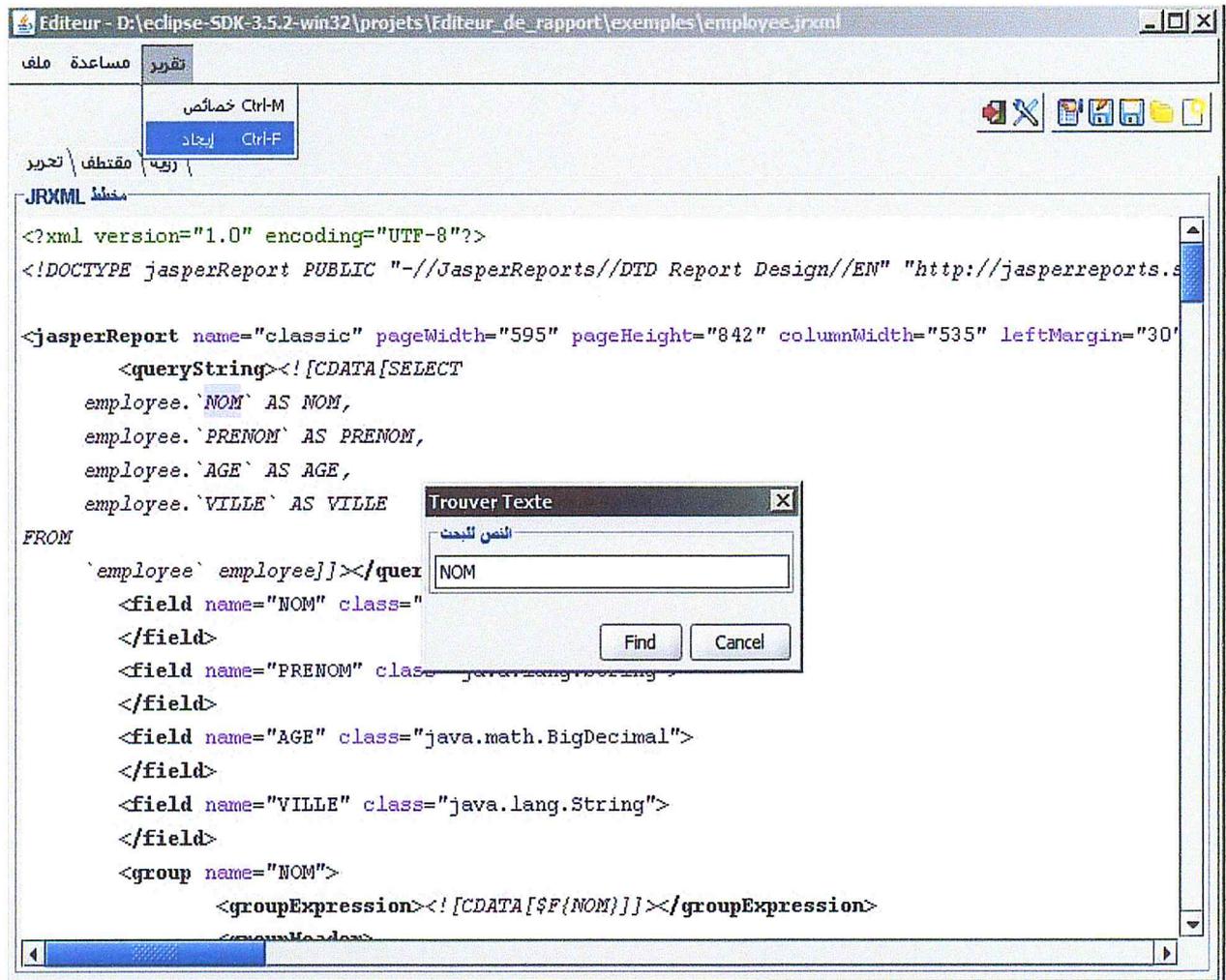


Figure.12. Recherche du texte

III.5.4 Préférences

L'application offre plusieurs aspects d'interface, l'utilisateur aura la possibilité de changer le thème qu'il souhaite en cliquant sur **تفضيلات** (figure.13) :

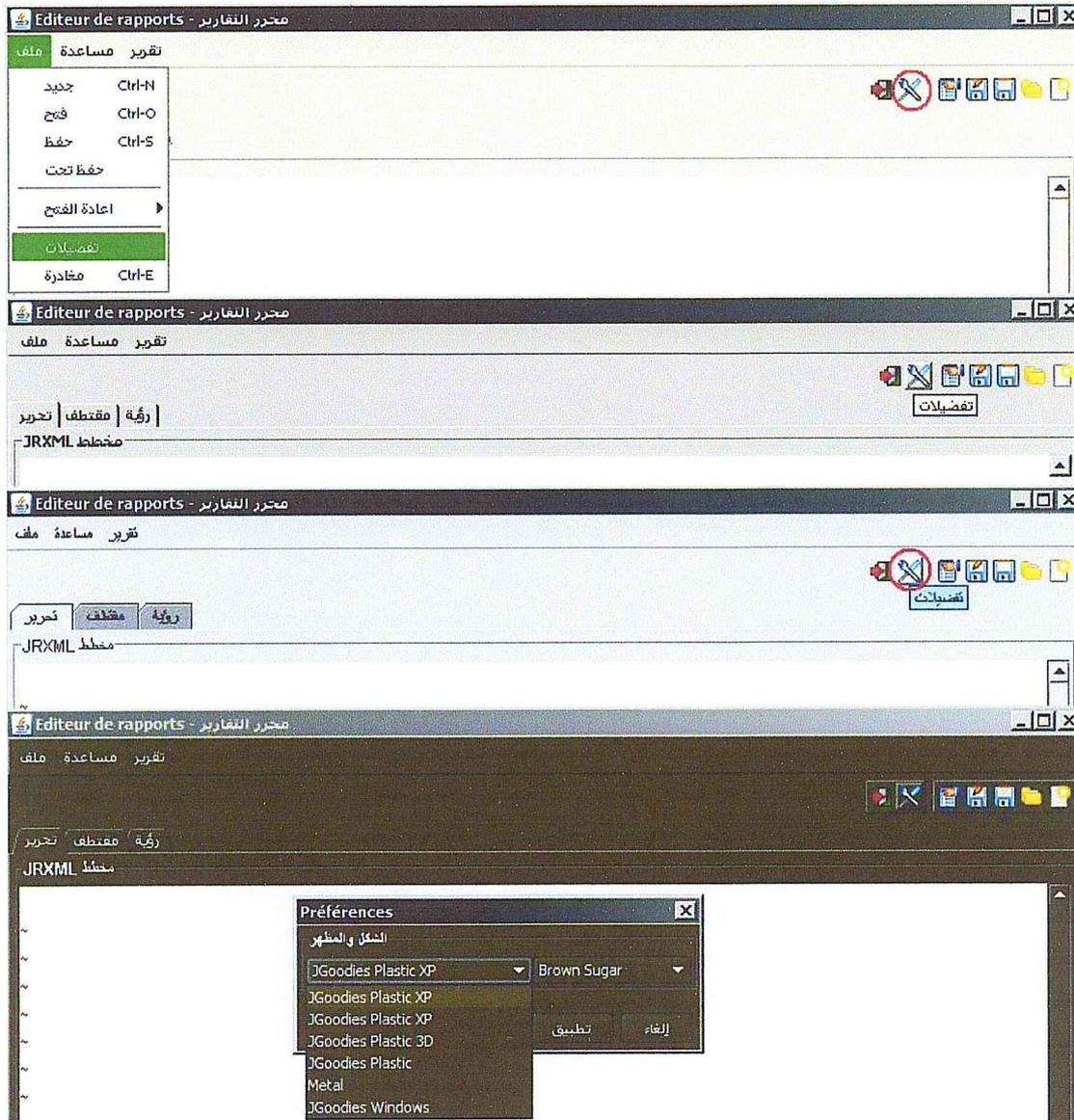


Figure.13. les différents thèmes de l'application

III.5.5 Propriétés de base du rapport

Pour accéder aux différentes propriétés du rapport en appuyant sur **Ctrl-M** **خصائص**, mais un fichier doit être chargé en premier. Cette option permet à l'utilisateur de modifier les attributs du fichier tel que : largeur, hauteur et marges de page... etc. (figure.14)

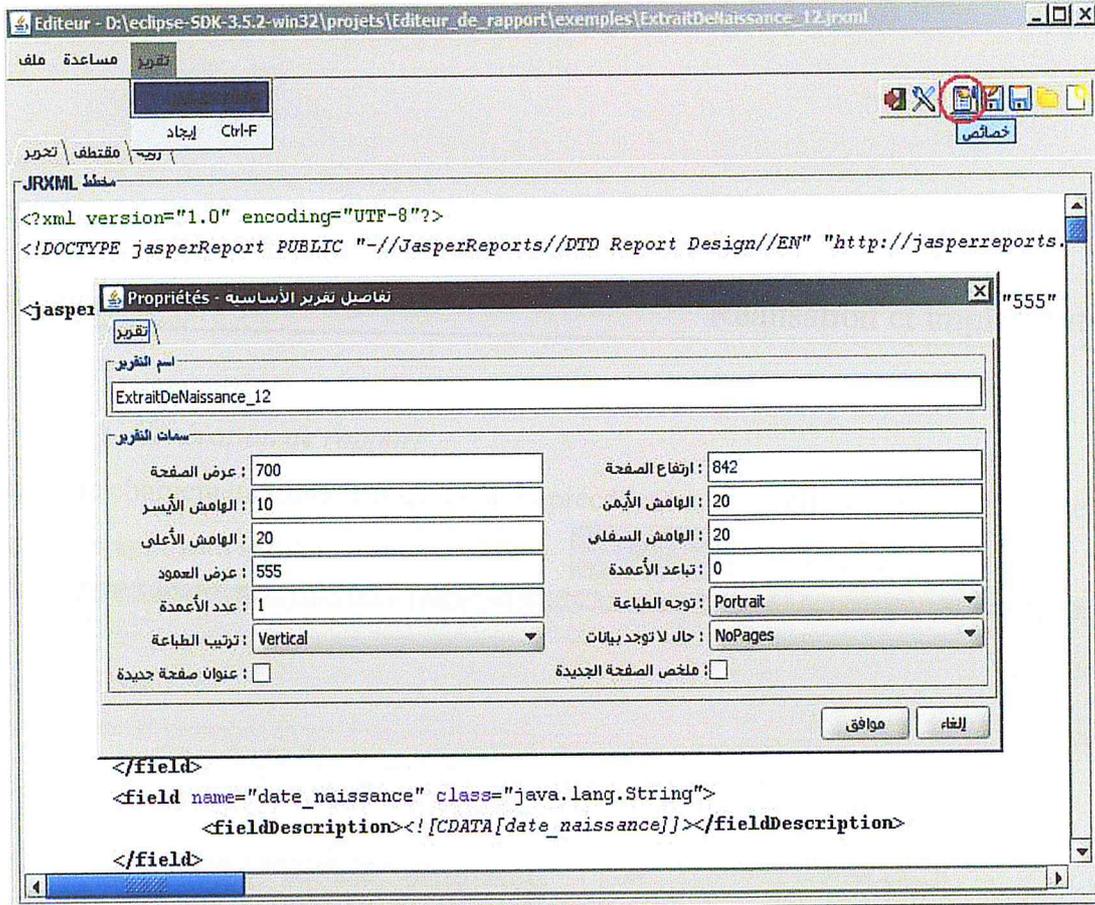


Figure.14. Propriétés

III.5.5.1 Les différents éléments de propriétés

Nous allons monter où les différents changements s'effectuent réellement dans le fichier JRXML.

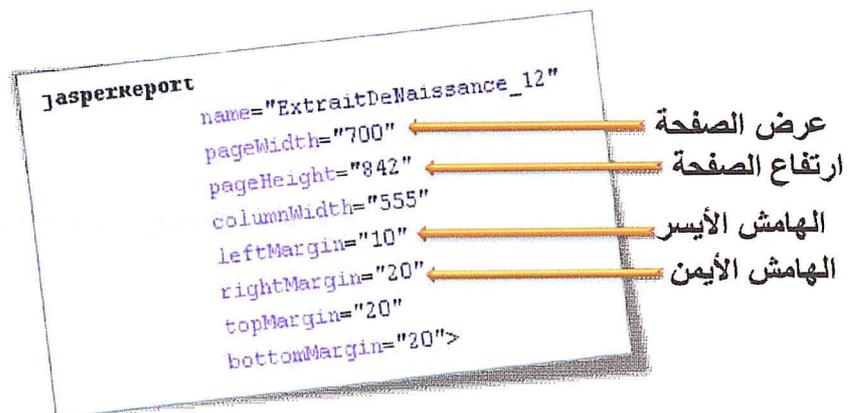


Figure.15. Les différents éléments de propriétés

III.6 Test de l'outil

Dans cette section, nous allons tester notre application comme démontré par les figures ci-dessous.

III. 6. 2 Aperçu du rapport d'impression

Pour visualiser le rapport d'impression (rapport final) il suffit juste d'appuyer sur l'onglet **روية**. La **figure 17** ci-dessous nous montre le décalage bien visible entre le texte et son champ.

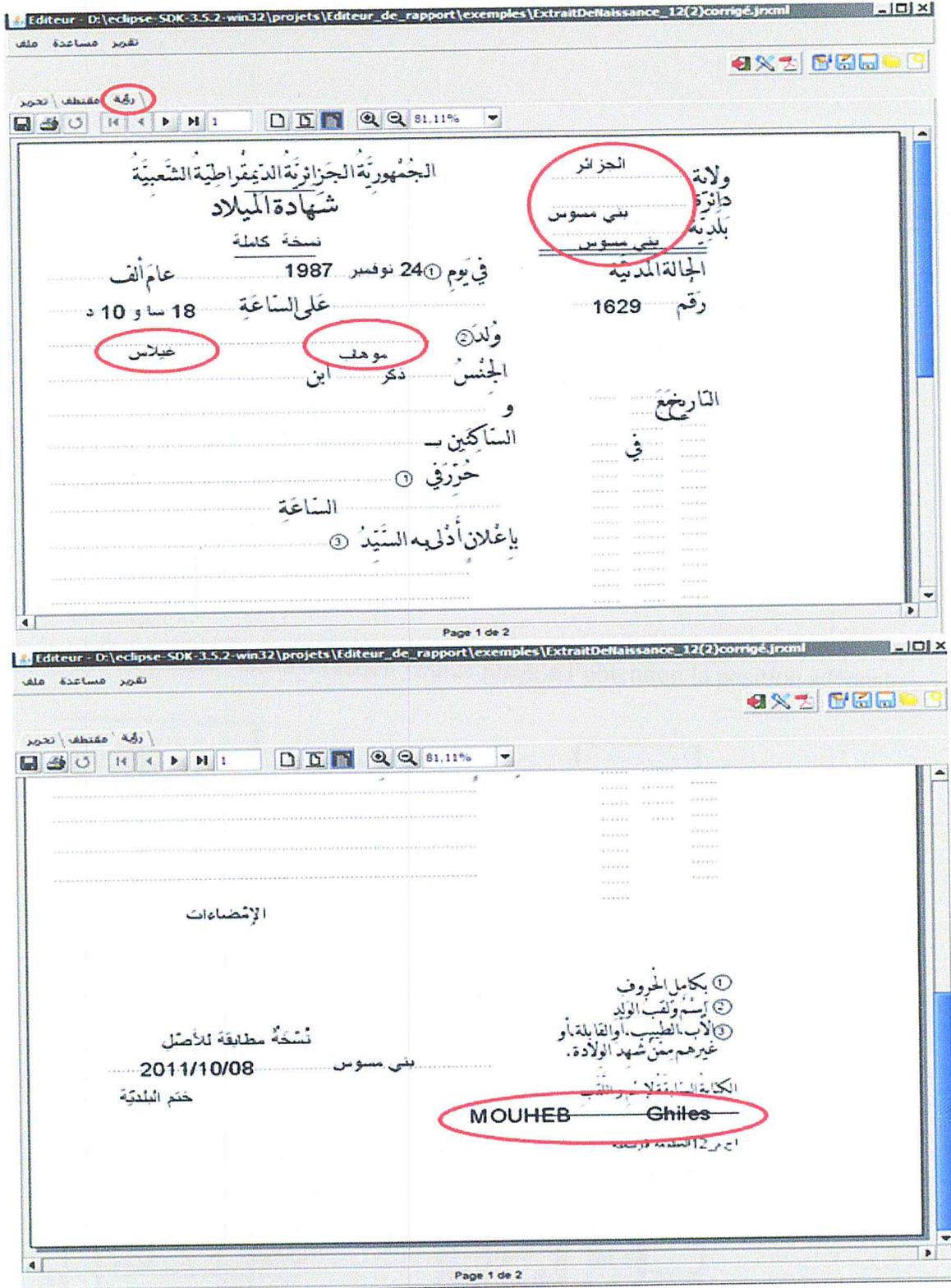


Figure.17. Aperçu du rapport final

III. 6. 4 Aperçu du rapport d'impression après rectifications (figure.19)

Avant de voir les modifications effectuées, une sauvegarde est nécessaire à l'aide de ce raccourci .

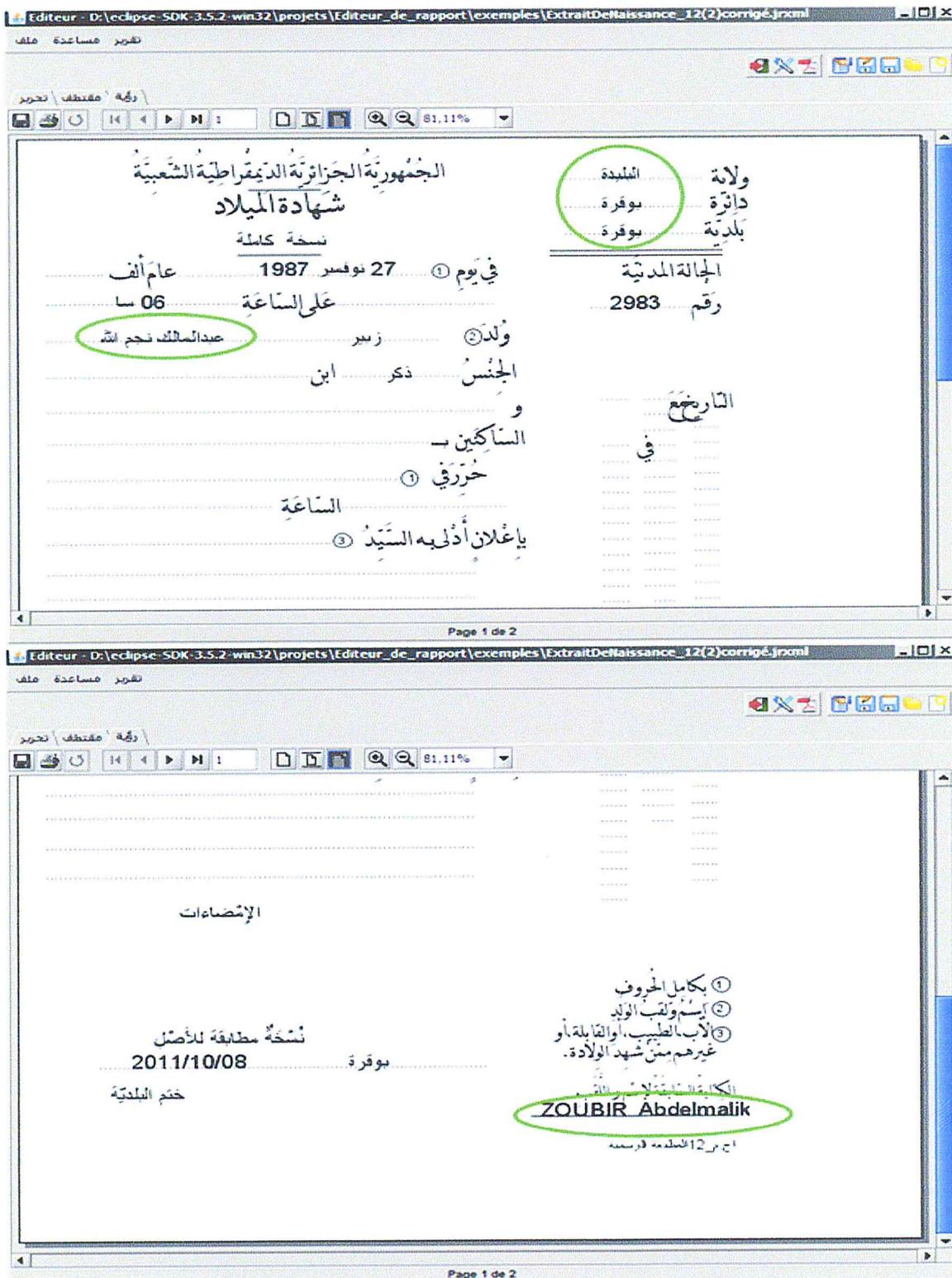


Figure.19. Aperçu du rapport final après rectifications

III. 6.5 Aperçu du rapport d'impression en format PDF

Le raccourci  permet d'exporter le rapport final au format PDF.

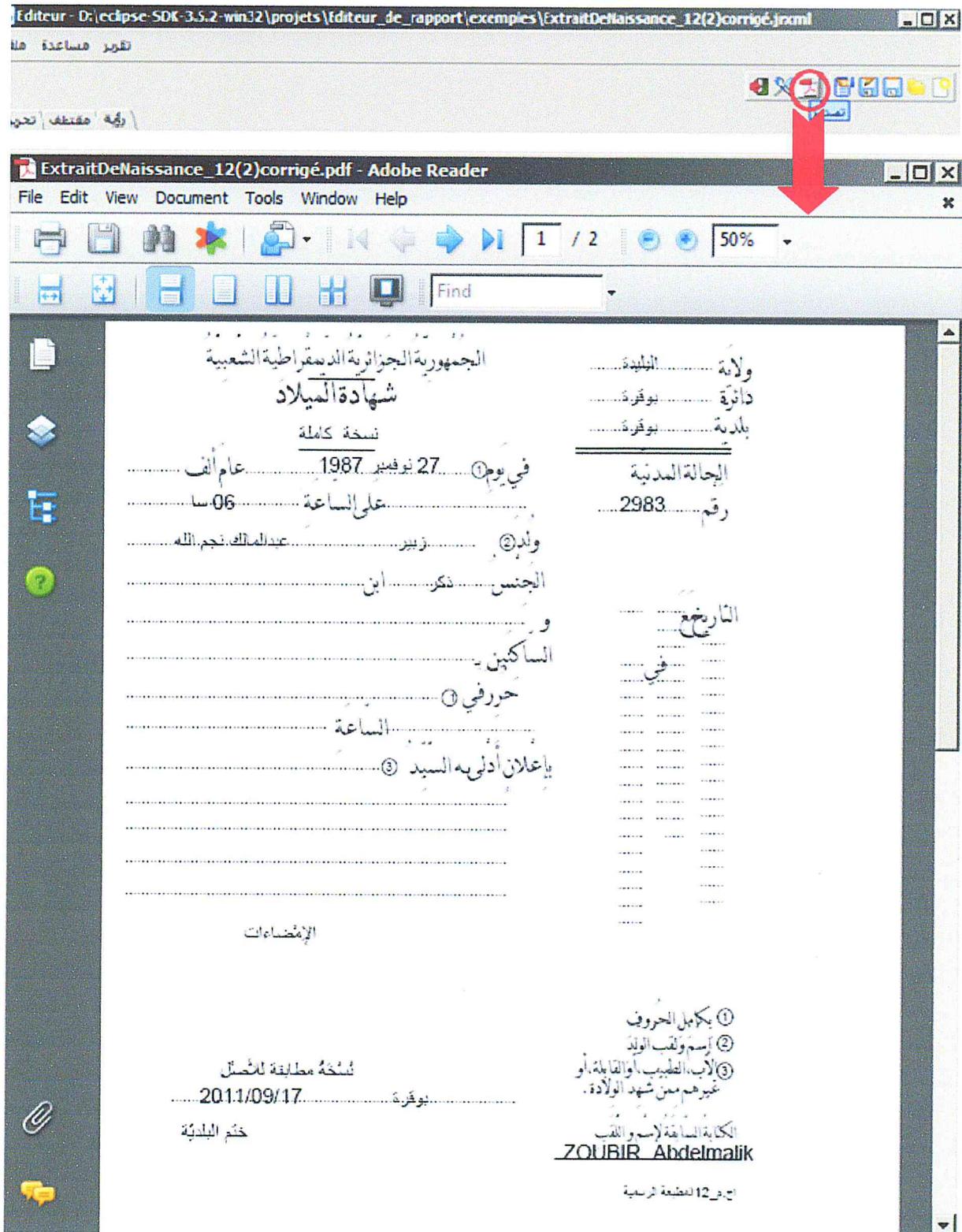
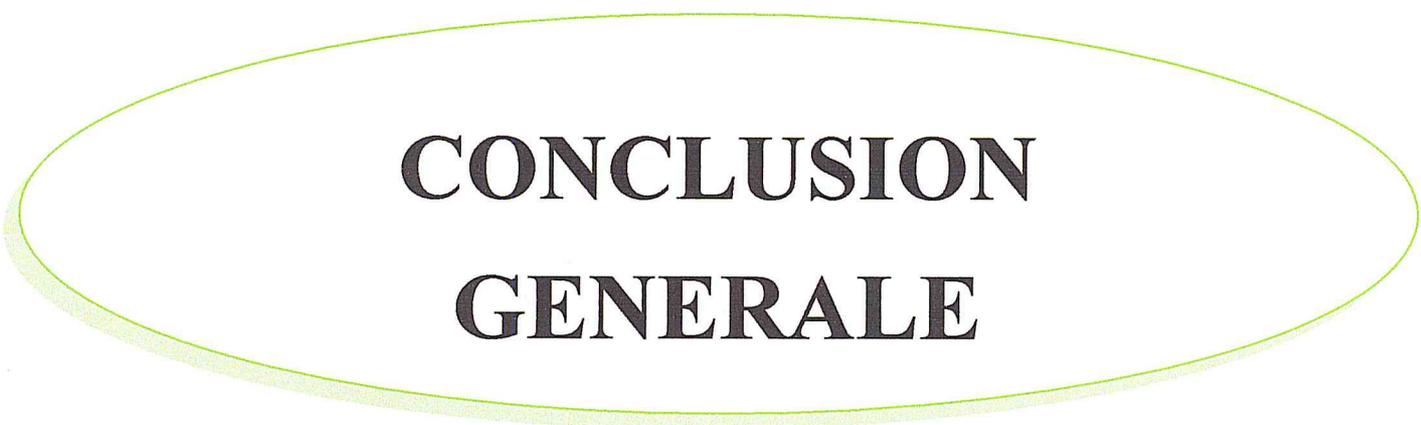


Figure.20. Exportation du rapport au format PDF



**CONCLUSION
GENERALE**

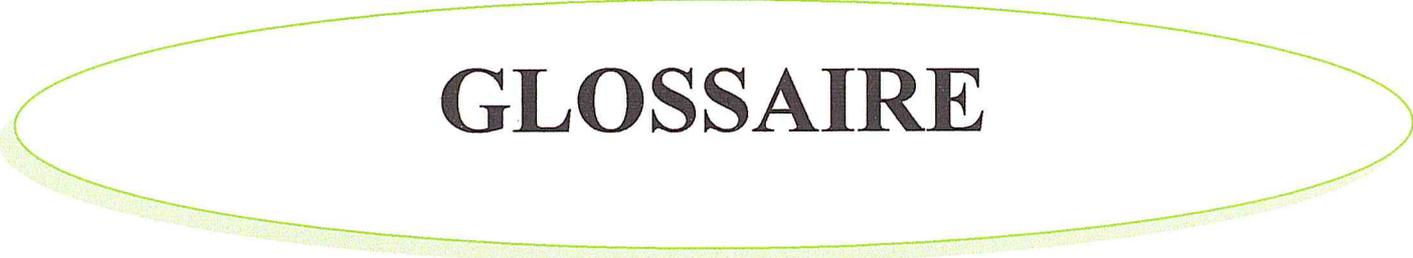
CONCLUSION

L'objectif de ce travail était de concevoir et de réaliser un outil de reporting, permettant à l'utilisateur la génération des rapports et leur manipulation d'une manière simple et facile.

Durant ce projet, nous avons étudié les différents outils de reporting existants, ce qui a permis de choisir les meilleures approches pour concevoir et développer une application qui permet de faire le reporting d'une manière très aisée pour l'utilisateur.

Ce projet nous a donné l'occasion d'acquérir de nouvelles connaissances dans le reporting, avoir de l'expérience en programmation java et se familiariser avec l'environnement java et les fichiers xml.

En dépit de certaines difficultés que nous avons rencontrées, dont la nécessité de se familiariser avec les outils de reporting et les documents xml, et ce qui prenait un temps considérable ; nous avons tout de même pu réaliser le travail qui nous a été confié et ainsi réussi à atteindre notre but. Cependant, il restera toujours des améliorations à apporter, pour une meilleure performance.



GLOSSAIRE

- 1- **Applications Programming Interface** : est un ensemble particulier de règles («code») et spécifications que les logiciels peuvent suivre pour communiquer entre eux. Il sert comme interface entre les différents logiciels et facilite leur interaction, semblable à la manière dont l'interface utilisateur facilite l'interaction entre les humains et les ordinateurs.
- 2- **eGovernment** : diminutif d'electronic government, connu aussi sous le nom de e-gov, digital government, online government, or connected government. Il est défini aussi comme étant les interactions numériques entre un gouvernement et les citoyens (G2C), le gouvernement et les entreprises /Commerce (G2B), le gouvernement et les employés, et aussi entre le gouvernement et les gouvernements / agences (G2G). Son rôle consiste à délivrer des informations et des services gouvernementaux aux citoyens.
- 3- **Java** : est un langage de programmation développé à l'origine par James Gosling chez Sun Microsystems. Il tire une grande partie de sa syntaxe de C et C++, mais possède un modèle objet simple et moins bas niveau d'installations. Les applications Java sont généralement compilées en bytecode (fichier de classe) qui peut fonctionner sur n'importe quelle machine virtuelle Java (JVM), indépendamment de l'architecture des ordinateurs.
- 4- **Librairie (JAR)** : le JAR (pour Java Archive) est un format de fichier d'archive généralement utilisé pour agréger plusieurs fichiers de classe Java et associés des métadonnées et des ressources (textes, images et ainsi de suite) dans un seul fichier à distribuer des logiciels d'application ou des bibliothèques sur la plate-forme Java. Les fichiers JAR sont construits sur le format de fichier ZIP et ont l'extension du fichier .jar. Les utilisateurs d'ordinateurs peuvent créer ou extraire des fichiers JAR à l'aide de la commande jar qui vient avec un JDK (Java Development Kit).
- 5- **W3C** : Le **World Wide Web Consortium**, abrégé par le sigle **W3C**, est un organisme de standardisation à but non-lucratif, fondé en octobre 1994 comme un consortium chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, CSS(Cascading Style Sheets) et PNG(Portable Network Graphics).

Références bibliographiques

1. **Anonyme.** (2004). XML. [en ligne] <http://www.startyourdev.com/xml/xml-presentation.html> . (consulté le 10.06.11)
2. **Anonyme.** (2005). Choix d'un outil de reporting. 4p.
3. **Atol conseil et développement.**(2005). Présentation de l'outil de reporting « JasperReports » [en ligne] <http://www.atolcd.com/onglets/recherche.html> (consulté le 25.05.11)
4. **Bigand M.** (2006). Conception des systèmes d'information: modélisation des données, études de cas. Editions TECHNIP. 165 p
5. **Brillant A.** (2007). XML : Cours Et Exercices. Eyrolles. 284 p
6. **Briol P.** (2008). Les Fondements de L'architecture D'entreprise. Patrice Briol. 209p
7. **Booch G. Rumbamgh, J. Jacobson, I.** (2003). Le guide de l'utilisation UML. 2° Ed. Eyrolles. 534 p.
8. **Danciu T. et Chirita L.** (2007). The Definitive Guide to JasperReports : Chapter7 Report Template Structure. Apress. 223 p
9. **Debrauwer L. et Van der Heyde F.** (2008). UML 2: Initiation, exemples et exercices corrigés. 2° Ed. .281 p
10. **Fernandez A.** (1998-2011). Reporting, les outils de publication de rapport. [en ligne] <http://www.piloter.org/business-intelligence/reporting.htm> (consulté le 30.05.11)
11. <http://oreports.com/> (consulté le 20.08.11)
12. **JauB.** (2009). FAC Jasper iRport. [en ligne]. 21p . < http://business-intelligence.developpez.com/faq/jasper-ireport/?page=general#GENERAL_what_is_jasperreports > (consulté le 25.05.11)
13. **Lecomte S. et Boulanger T.** (2008). XML par la pratique: bases indispensables, concepts et cas pratiques. 2e Ed. Editions ENI. 347 p
14. **Laugié H.** (2008). Java et Eclipse: développez une application Java. Ediciones ENI. 268 p.
15. **Muller P.A. et Gaertner N.** (2005). Modélisation objet avec UML . 2° Ed. Eyrolles. 514 p.
16. **Miller F.P., Vandome A. F., McBrewster J.** (2010). Document Type Definition. VDM Publishing House Ltd. 96 p
17. **Toffoli, G.** (2007). The Definitive Guide to IReport. Apress. 352 p