

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Gestion de Parc Informatique Matériel et Logiciel

Réalisé par :

- Mr DJABEUR DJEZZAR Mohammed Rafik
- Mr MERABET Aissa

Présenté le 02 Juin 2015 devant la commission d'examination composée de MM.

- Mr SMAHI Mohammed Ismail (Encadreur)
- Mr MERZOUGUE Mohammed (Examineur)
- Mme YABDERI Zineb (Examineur)

Année universitaire : 2014-2015

Dédicaces

*Au nom du dieu clément et miséricordieux
Je dédie ce travail à :*

*Mes chères parents qui n'ont jamais cessé de m'encourager tous
le long de mon parcours et qui ont toujours tout sacrifié pour faire
de moi ce que je suis à présent que dieu les protège.*

Tous mes amis et mes amies.

Ceux qui me sont proches et chers de loin ou de près.

Rafik

Dédicaces

*Au nom du dieu clément et miséricordieux
Je dédie ce travail à :*

*Mes chères parents qui n'ont jamais cessé de m'encourager tous
le long de mon parcours et qui ont toujours tout sacrifié pour faire
de moi ce que je suis à présent que dieu les protège.*

*Et ma femme et mon fils MOHAMMED RIADH.
A la lumière de mes jours, la source de mes efforts
Ma femme et mon fils.*

Tous mes amis et mes amies.

Ceux qui me sont proches et chers de loin ou de près.

Aissa

Remerciements

Ce mémoire, ne pourrait exister sans l'aide et l'engagement d'un certain nombre de personnes qui ont décidé de nous accompagner résolument dans notre parcours.

Que tous ceux qui, de près ou de loin, ont contribué par leurs conseils, leurs encouragements et leur assistance à l'aboutissement de ce travail, trouvent ici l'expression de notre profonde gratitude.

Tout d'abord, nous tenons à remercier nos encadreurs : M. SMAHI Mohammed Ismail pour les efforts qu'il a fourni pour nous aider et nous orienter dans notre travail jusqu'à sa finalisation.

Nous remercions, Mr MERZOUGUE Mohammed, Mme YABDERI Zineb d'avoir accepté de faire partie de ce jury, nous en sommes très honorés.

Merci à nos proches et amis de nous soutenir par leur présence dans les bons comme dans les mauvais moments.

Merci à nos enseignants pour la qualité et le dévouement à la cause éducative. Qu'ils trouvent ici l'expression de toute notre reconnaissance.

Liste des figures

Figure 1 : l'organigramme de la société OPGI	13
Figure 2 : Diagramme de cas d'utilisation	18
Figure 3 : Diagramme de classes	19
Figure 4 : Diagramme d'états-transitions	19
Figure 5 : Diagramme de séquence.....	20
Figure 6 : Diagramme de collaboration	20
Figure 7 : Diagramme d'activités	20
Figure 8 : Diagramme de déploiement	21
Figure 9 : Notation UML de la classe Point	22
Figure 10 : Symbole de l'association	22
Figure 11 : Symbole de l'agrégation	22
Figure 12 : Symbole de la composition	22
Figure 13 : Symbole de la généralisation.....	23
Figure 14 : Notation UML d'une transition entre deux états	23
Figure 15 : Expression de la contrainte n 1 en OCL.....	25
Figure 16 : Expression de la contrainte n 2 en OCL.....	25
Figure 17 : Expression de la contrainte n 3 en OCL.....	25
Figure 18 : Expression de la contrainte n 4 en OCL.....	25
Figure 19 : Diagramme de cas d'utilisation.....	29
Figure 20 : Diagramme de séquence « authentification »	30
Figure 21 : Diagramme de séquence « Affectation matériel »	31
Figure 22 : Diagramme de séquence « Installation_Déstallation »	31
Figure 23 : Diagramme de séquence « Intervention »	32
Figure 24 : Diagramme de Classe.....	33
Figure 25 : Fiche d'authentification.....	37
Figure 26 : Fiche principale	38
Figure 27 : Fiche Matériel	38
Figure 28 : Fiche Fournisseur	39
Figure 29 : Fiche Intervention	39
Figure 30 : Fiche Intervenant.....	40
Figure 31 : Fiche Marque.....	40
Figure 32 : Fiche Software.....	41
Figure 33 : Fiche Département	41
Figure 34 : Fiche Structure	42

Table des matières

Dédicaces	1
Dédicaces	2
Remerciements.....	3
Liste des figures	4
Introduction Générale	7
Introduction générale :	8
Chapitre I : Etude de L’Existant	10
I. INTRODUCTION :	11
II. PRESENTATION DE L’ORGANISME D’ACCEUIL :	11
III. HISTORIQUE DE L’OPGI :	11
IV. MISSION DE L’OFFICE :	11
L’Office de Promotion et de Gestion Immobilière de Tlemcen a pour missions principales :	11
V. ORGANIGRAMME DE L’ORGANISME :	12
VI. LA CELLULE O.S.I.C :	14
A. Organisation et système d’information.....	15
1) Gestionnaire du parc micro informatique/ Maintenance.....	15
Principales taches :	15
2) Analyste Développeur	15
3) Administrateur des Bases de données.	15
B. Réseaux et communication	16
V. Administrateur du site web : (Webmaster)	16
VI. Administrateur du réseau :	16
Chapitre II : Le langage UML	17
Historique.....	18
I. LA NOTATION GRAPHIQUE :	18
II. LES DIFFERENTS DIAGRAMMES	18
1) Diagrammes de cas d'utilisation :	18
2) Diagrammes de classes :	19
VII. Diagrammes d'objets :.....	19
VIII. Diagrammes d'états-transitions :.....	19
IX. Diagrammes de séquence :	19
X. Diagrammes de collaboration :.....	20

XI.	Diagrammes d'activités :	20
XII.	Diagrammes de composants :	21
XIII.	Diagrammes de déploiement :	21
XIV.	Diagrammes de classes :	21
III.	Description d'une classe :	21
VII.	Relations entre classes :	22
V.	DIAGRAMME D'ETATS-TRANSITIONS	23
XV.	OCL LE LANGAGE DE CONTRAINTES	24
1.	Origine	24
2.	Généralités	24
3.	Exemples de contraintes <i>OCL</i> .	25
V.	CONCLUSION :	26
Chapitre III : Modélisation du système		27
I.	Introduction :	28
II.	Environnement de modélisation :	28
III.	Modélisation de système :	28
IV.	Les diagrammes de séquence :	30
1)	Authentification :	30
2)	Affectation Matériel :	31
3)	Installation _ Désinstallation :	31
4)	Intervention :	32
V.	Diagramme de Classe :	32
VI.	Modèle Logique des Données :	34
VII.	Conclusion :	34
Chapitre IV : Implémentation du Système		35
I.	Introduction :	36
II.	L'environnement de programmation Microsoft Visual Studio 2010 sp1 :	36
III.	L'environnement de gestion de base de donnée :	36
IV.	Description de l'application :	36
V.	Conclusion :	42
Conclusion générale		43
Bibliographique		44
Annexes		45

Introduction Générale

Introduction Générale

Introduction générale :

La plupart des organisations possèdent aujourd'hui un réseau d'ordinateurs privé. Au travers de ce réseau, les postes échangent des fichiers, partagent de matériel et parfois utilisent des applications (logiciels) en commun.

De nos jours, la plupart des applications de gestion ont une architecture client-serveur. Ces applications constituent le cœur de ce que l'on appelle le système d'information central.

Ces applications gèrent des plannings, des annuaires, des ressources, des stocks, des commandes, des livraisons, etc. Ce qui permet de mettre facilement à la disposition des employés des documents divers et variés et oblige un accès centralisé à la mémoire de l'entreprise. Il est donc généralement nécessaire de définir des droits d'accès pour les utilisateurs de l'intranet et par conséquent une authentification de ceux-ci afin de leur permettre un accès personnalisé aux fonctionnalités assurées par l'intranet.

De cette façon, un intranet favorise la communication au sein de l'entreprise et limite les erreurs dues à la mauvaise circulation d'une information.

Au sein de la société OPGI, notre mission était de choisir un portail répondant à ses exigences et le mettre en place. L'application de la Gestion de parc informatique après avoir été conçue et développée doit s'intégrer dans ce portail avec l'ensemble des applications du système d'information de la société. En effet, La gestion de parc permet de suivre en temps réel du patrimoine informatique, matériel et logiciel de l'entreprise.

Elle offre une vision globale de l'état, du suivi et des coûts des appareils utilisés dans l'entreprise et de bien Gérer les différents types d'équipements (Unités Centrales, Ecrans, Imprimantes, Matériels Réseaux, ... ext) ainsi que leurs Composants Hard et Soft (Processeurs, Mémoires, Disques durs, OS, Logiciels...) et aussi visant à assurer le bon fonctionnement des PC et serveur.

Dans cet objectif, notre travail consiste à réaliser une application de gestion de parc informatique. Pour la mise en œuvre de notre application nous avons utilisé l'environnement Model Maker comme outils de modélisation en langage **UML**, ainsi que l'environnement Microsoft Visual Studio pour la réalisation de l'application.

Le présent rapport décrit les différentes étapes de la réalisation du projet. Il comporte quatre chapitres :

- ❖ Le premier chapitre est consacré à l'Etude de l'Existant
- ❖ Le deuxième chapitre est consacré à la présentation du langage **UML** (Le définition, les descriptions, et l'utilité de différents diagrammes).

Introduction Générale

- ❖ Le troisième chapitre est consacré à la modélisation du système étudié, on utilisant les différents diagrammes (diagramme de cas d'utilisation, diagramme de séquence et diagramme de classe) d'**UML**.
- ❖ Le dernier chapitre consiste à la phase de réalisation et de mise en œuvre de l'application.
- ❖ Finalement, nous terminons notre mémoire par une conclusion générale contenant quelques perspectives pour le système de gestion d'un parc informatique.

Chapitre I : Etude de L'Existant

I. INTRODUCTION :

L'analyse de l'existant est une étape importante dans le cycle de vie d'un système, il s'agit de connaître la situation actuelle de l'organisation pour pouvoir porter un jugement juste. Ainsi, l'analyse du système existant doit nous fournir toute l'information nécessaire, afin d'établir une bonne conception et de proposer de bonnes solutions.

Dans ce chapitre, nous allons présenter une étude Générale sur l'Office de Promotion et de Gestion Immobilière de Tlemcen «O.P.G.I». Par la suite, un intérêt particulier est porté à la cellule Informatique O.S.I.C «Organisation et Système d'Information et de Communication ».

II. PRESENTATION DE L'ORGANISME D'ACCEUIL :

- L'Office de Promotion et de Gestion Immobilière de TLEMCEM, par abréviation «OPGI» est une entreprise publique à caractère industrielle et commerciale.
- L'Office est une personne morale, jouissant de l'autonomie financière.
- L'Office est géré par un conseil d'administration dont les membres sont désignés par le ministère de tutelle.
- L'Office est dirigé par un directeur général, nommé par décret présidentiel.

III. HISTORIQUE DE L'OPGI :

➤ Créé par ordonnance N°74-143 du 23 Octobre 1976 portant sur la création des OPGI <<Office de Promotion et de Gestion Immobilière>> de WILAYA, l'OPGI n'a cessé depuis de subir de réformes et des transformations de nature juridique et organisationnelle suivant l'évolution des besoins politiques et des programmes nationaux dans différents contextes et ceci dans le but de faciliter la prise en charge effective des attributions de l'office en matière de maîtrise d'ouvrage des réalisations de logement sociaux [1].

➤ En effet d'un établissement public local sous tutelle de la WILAYA, l'OPGI de TLEMCEM est passé de statut d'établissement public national à caractère industriel et commerciale, régi par les dispositions du décret N°91-147 du Mai 1991 portant sur la transformation de la nature juridique des OPGI et les modalités de leur organisation et de leur fonctionnement, doté de personnalité morale et de l'autonomie financière dont la tutelle est exercée par le ministère de l'habitat et de l'urbanisme [2].

IV. MISSION DE L'OFFICE :

L'Office de Promotion et de Gestion Immobilière de Tlemcen a pour missions principales :

- ✚ De promouvoir le service public en matière de logement social.
- ✚ La gestion immobilière.

Il est chargé à titre accessoire de :

- ✚ La promotion immobilière.
- ✚ La maîtrise d'ouvrage déléguée pour le compte de tout autre opérateur.
- ✚ Actions de prestation de services en vue d'assurer l'entretien, la maintenance, la réhabilitation et la restauration des biens immobiliers.
- ✚ La promotion foncière.
- ✚ Toutes actions entrant dans la gestion immobilière.

V. ORGANIGRAMME DE L'ORGANISME :

L'organigramme représente graphiquement la structure de l'office « **OPGI** », cet organigramme appartient à la forme traditionnelle et en particulier à la structure hiérarchique, en effet chaque individu n'a qu'un seul supérieur.

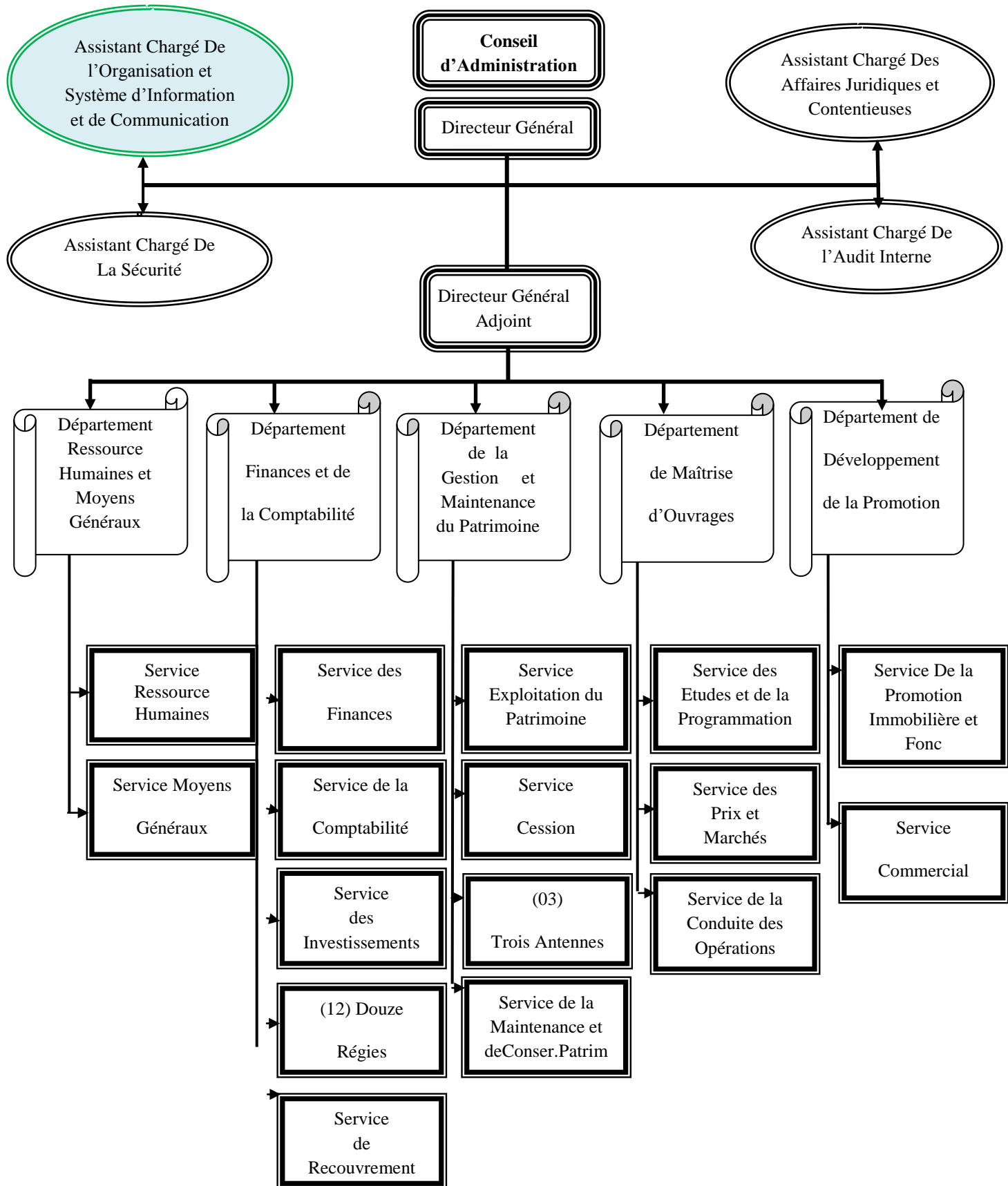


Figure 1 : l'organigramme de la société OPGI

VI. LA CELLULE O.S.I.C :

Objectif

- ▶ Délimitation du domaine d'intervention de la cellule.
- ▶ Identification des moyens pour l'accomplissement des missions confiées à la cellule.

La cellule O.S.I.C faisant partie du staff de la direction, sous l'autorité du Directeur Général, conformément à l'organigramme approuvé par la décision N° 218/DRHR/06.

principales missions :

- ▶ Faire évoluer le système d'information.
- ▶ Mettre en œuvre les solutions informatiques.
- ▶ Gérer les investissements Informatiques.
- ▶ Administrer les bases de données et les réseaux.
- ▶ Gérer du site web de l'Office.
- ▶ Assurer La veille technologique.

Organisation

A. Organisation et système d'information

1. Gestionnaire du parc micro informatique/ Maintenance
2. Analyste Développeur
3. Administrateur des bases de données

B. Réseaux et communication

4. Webmaster / Administrateur du site web
5. Administrateur des Réseaux

A. Organisation et système d'information

1) Gestionnaire du parc micro informatique/ Maintenance

Principales taches :

- ▶ Maintenir, upgrader, dépanner et veiller sur le parc micro-informatique.
- ▶ Assistance aux utilisateurs.

2) Analyste Développeur

Fusion de deux métiers en informatique, Architecte du système d'information, et le développeur (programmeur).

Principales taches sont :

- ▶ Le recueil de l'information
- ▶ La Modélisation du système d'information
- ▶ Propositions et conception de la nouvelle solution.
- ▶ Réalisation de la nouvelle solution.
- ▶ Rédiger le manuel des logiciels informatiques
- ▶ Assurer la maintenance des logiciels.

3) Administrateur des Bases de données.

Principales taches :

- ▶ Définition des structures des données.
- ▶ Assuré la cohérence des données.
- ▶ La réorganisation permanente pour réduire le temps de réponse.
- ▶ Le maintien et l'évolution de la base de données
- ▶ La définition des privilèges. (Gestion des Utilisateurs)
- ▶ Assurer la sécurité de la base de données
- ▶ Interrogation de la base de données.
- ▶ Suivi des Mises a jour du fichier National.

B. Réseaux et communication

V. Administrateur du site web : (Webmaster)

Principales missions:

- ▶ La conception et la réalisation du site web.
- ▶ Evolution permanente du site web.
- ▶ Mise a jour du site web.
- ▶ Gestion du centre d'écoute et réponse aux e-mails.

VI. Administrateur du réseau :

Principales missions :

- ▶ La mise en œuvre, la gestion et l'optimisation des réseaux de communication internes et externes.
- ▶ Choix des équipements logiciels et matériels en réseaux et télécommunications.
- ▶ La conception, la mise en œuvre et le suivi des différents réseaux et de leurs interconnexions.
- ▶ Collabore avec l'analyste développeur pour rédiger des spécifications techniques lors de la mise en œuvre de nouvelles solutions.

Chapitre II : Le langage UML

Historique

Avec le succès grandissant de la programmation orientée objet, de nombreuses méthodes semi-formelles d'analyse et de conception ont été dédiées à ce nouveau paradigme. **UML** (Unified Modeling Language) est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au début des années 90: OMT, OOAD et OOSE Le projet, initié par Rumbaugh, Booch et Jacobson au sein de la société Rational, devient en 1997 une norme de l'Object Management Group [3].

I. LA NOTATION GRAPHIQUE :

La notation **UML** se répartit en différents types de diagrammes, qui décrivent des aspects complémentaires mais non disjoints du système modélisé. Nous décrirons plus particulièrement les diagrammes de classes et les diagrammes d'états-transitions, qui sont les deux diagrammes pris en compte lors de la conversion d'**UML** vers B.

II. LES DIFFERENTS DIAGRAMMES

UML définit neuf sortes de diagrammes. La plupart d'entre eux se présentent sous la forme de graphes, composés de sommets et d'arêtes. Les exemples qui suivent ont été réalisés à l'aide du logiciel **ArgoUML** version 0.14.1.

1) Diagrammes de cas d'utilisation :

Représentent les fonctions du système du point de vue de l'utilisateur. Un cas d'utilisation décrit la réponse du système à un événement provenant d'un acteur. Les scénarios sont des instances de cas d'utilisation.

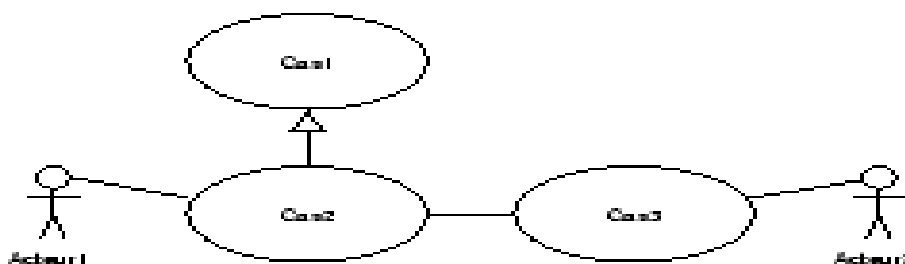


Figure 2 : Diagramme de cas d'utilisation

2) Diagrammes de classes :

Matérialisent la structure des données en termes de classes et de relations entre ces classes.

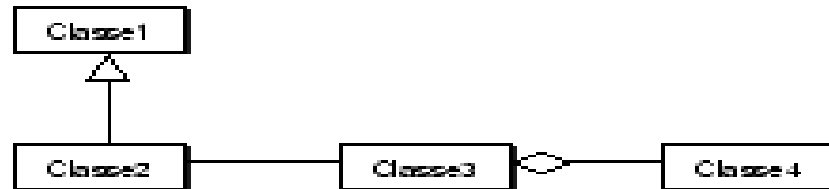


Figure 3 : Diagramme de classes

VII. Diagrammes d'objets :

Mettent en scène les objets et leurs relations : ils correspondent à des instances des éléments qui apparaissent dans les diagrammes de classes. Ils sont similaires aux diagrammes de collaboration, mais sans représentation des envois de messages.

VIII. Diagrammes d'états-transitions :

Matérialisent la vue dynamique du système. Ils décrivent le comportement d'une classe en terme de changements d'états.

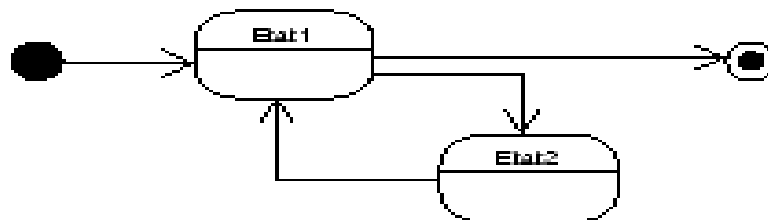


Figure 4 : Diagramme d'états-transitions

IX. Diagrammes de séquence :

Proposent une représentation temporelle des objets et de leurs interactions. Les scénarios des diagrammes de cas d'utilisation constituent une version simplifiée de diagrammes de séquence.

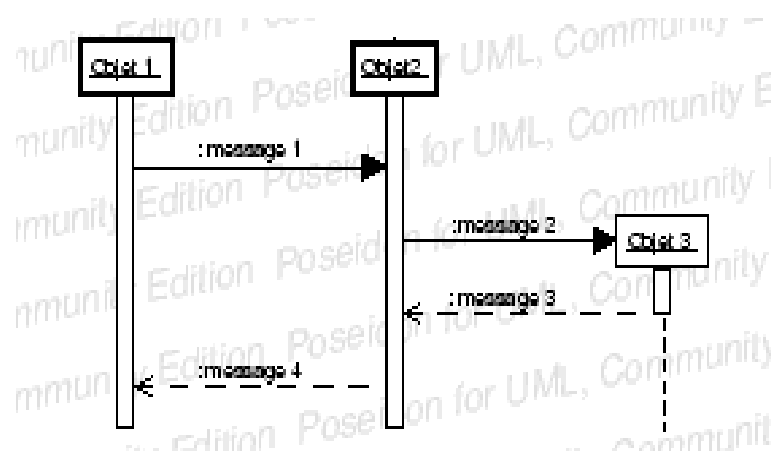


Figure 5 : Diagramme de séquence

X. Diagrammes de collaboration :

Sont une représentation spatiale des objets, des liens et des interactions. Ils contiennent les mêmes informations que les diagrammes de séquence, mais soulignent l'organisation structurelle des objets qui envoient et reçoivent des messages.

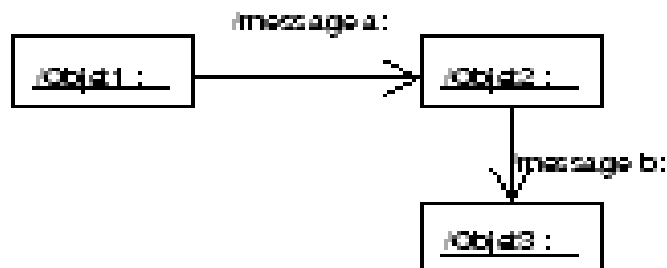


Figure 6 : Diagramme de collaboration

XI. Diagrammes d'activités :

Décrivent le comportement d'une opération en termes d'actions. C'est la représentation duale du diagramme d'états-transitions, les états correspondant à des activités et les transitions à des fins d'activités.



Figure 7 : Diagramme d'activités

XII. Diagrammes de composants :

Mettent en scène les composants physiques d'une application : ils mettent en évidence la structure logicielle du système. Chaque composant est une partie de l'implémentation du système.

XIII. Diagrammes de déploiement :

Décrivent le déploiement des composants sur les dispositifs matériels. Ce sont des graphes, dont les sommets sont des ressources de calcul (processeur ou périphérique) et les arêtes des supports de communication (réseau, par exemple).

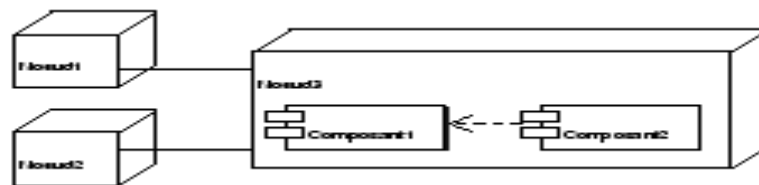


Figure 8 : Diagramme de déploiement

XIV. Diagrammes de classes :

Un diagramme de classes est une collection d'éléments modélisant la structure du système en faisant abstraction des aspects dynamiques et temporels. C'est à la fois l'axe essentiel de la modélisation objet et la notation la plus riche de tous les diagrammes UML.

III. Description d'une classe :

Une classe est caractérisée par :

- son nom
- la liste de ses attributs (équivalents des variables d'instance de la programmation orientée objet)
- la liste de ses opérations (analogues aux méthodes)

Les attributs doivent être nommés et typés ; idem pour les opérations, leurs paramètres et les valeurs retournées. Un type d'attribut ou de paramètre d'opération est :

- soit un type de donnée, simple (entier, booléen, chaîne de caractères, ...) ou complexe (séquence, tuple, etc.).

- soit un type objet, dans le cas où il référence une autre classe du diagramme [5].

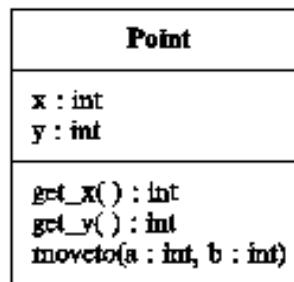


Figure 9 : Notation UML de la classe Point

VII. Relations entre classes :

Les quatre principaux types de relation entre classes sont :

- l'association, qui représente le lien unissant les instances des classes, un lien étant une relation d'utilisation (une connexion) entre différents objets.



Figure 10 : Symbole de l'association

- L'agrégation, qui est une forme d'association permettant d'exprimer une relation d'inclusion entre les classes (relation composant-composé).



Figure 11 : Symbole de l'agrégation

- La composition : c'est un cas particulier d'agrégation dans laquelle l'objet composé ne peut appartenir qu'à un seul objet composant (en termes de cycle de vie, les objets composés sont créés et détruits en même temps que l'objet composant).



Figure 12 : Symbole de la composition

- La généralisation. On peut spécialiser une super-classe, en ajoutant à la sous-classe des attributs et des opérations (c'est l'équivalent du concept d'héritage en programmation orientée objet). On dit alors que la super-classe est une généralisation de la sous-classe.

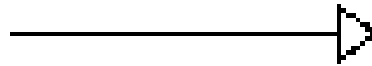


Figure 13 : Symbole de la généralisation

Chaque relation entre un couple de classes peut être dotée de rôles et de multiplicités (cf. le modèle Entité-Association). Un rôle sert à identifier de façon unique une extrémité d'association. Une multiplicité est associée à chaque extrémité pour indiquer, à l'aide d'un intervalle (0..1, 1..1, 0..* ou 1..*, * symbolisant l'infini), le nombre d'objets susceptibles de participer à une association donnée.

V. DIAGRAMME D'ETATS-TRANSITIONS

Un diagramme d'états-transitions décrit le comportement dynamique d'une classe donnée. La notation est proche de celle des statecharts de Harel : il s'agit de représenter des automates d'états finis sous la forme de graphes orientés, dans lesquels les sommets sont des états et les arcs des transitions entre les états. Un état représente une étape dans la vie d'un objet. Il se caractérise par sa durée et sa stabilité. Une transition symbolise un changement d'état instantané, induit par l'occurrence d'un événement. En plus de la description de l'événement, la transition peut être accompagnée d'une garde, condition sous laquelle on autorise le déclenchement de la transition, et/ou d'une action qui sera exécutée à l'entrée du nouvel état.



Signification :

- Si la classe est à l'état 1, elle y reste jusqu'à l'apparition de l'événement.
- Lorsque l'événement est déclenché :
 - * si la garde est satisfaite, la classe passe à l'état 2 et, au même moment, l'action est exécutée;
 - * sinon, la classe reste à l'état 1.

Figure 14 : Notation UML d'une transition entre deux états

XV. OCL LE LANGAGE DE CONTRAINTES**1. Origine**

OCL (Object Constraint Language) a été créé dans le but d'ajouter aux diagrammes de classes les informations qui ne pouvaient être exprimées par la notation graphique d'**UML**. C'est un langage de modélisation (ce n'est pas un langage de programmation) qui permet de spécifier formellement les contraintes associées aux objets d'un système et à leurs relations. Il est relativement simple d'accès, à mi-chemin entre langage naturel et langage mathématique. Ce langage objet, développé par **IBM**, a été intégré à **UML** dès sa version 1.1. Il a été utilisé notamment dans la définition du méta-modèle d'**UML**.

2. Généralités

En OCL, il existe trois types d'assertions : les pré-conditions, les post-conditions et les invariants [4].

- Une pré-condition (on utilise le mot-clé **pre**) permet de préciser les conditions sous lesquelles l'appel d'une opération aboutira à un comportement correct. C'est à l'objet qui invoque l'opération de vérifier ces conditions.

- Une post-condition (**post**) est un énoncé de ce que doit être le système après l'exécution d'une opération. C'est une façon d'exprimer ce que réalise l'opération sans dire comment elle procède.

- Un invariant (**inv**) est une assertion à propos d'une classe qui doit être vérifiée avant et après tout appel d'opération, quelle que soit l'instance de la classe. L'invariant peut être faux temporairement, pendant l'exécution d'une opération.

Les propriétés sont exprimées à l'aide d'opérations sur les collections, d'expressions de navigation ou d'expressions de requête :

- Les opérations sur les collections regroupent toutes les opérations classiques sur des ensembles : union, intersection, size, is Empty, etc... On utilise une flèche pour les invoquer.

- Les expressions de navigation permettent de référencer un attribut ou une association d'instance de classe (à l'aide de la notation pointée). Dans ce dernier cas, l'objet cible est accédé en utilisant le nom du rôle ou de la classe situé à l'extrémité de l'association.

- Les expressions de requête sont des opérations (notation fléchée) permettant de sélectionner des sous-ensembles d'objets (cf. le langage **SQL** pour les bases de données). Les six opérations de base sont: **select**, **reject**, **collect**, **forAll**, **exists** et **iterate**.

3. Exemples de contraintes OCL

Spécifions quelques contraintes sur notre exemple de la classe Point. Tout point doit avoir une abscisse et une ordonnée positives.

```
context Point inv :  
  self.x > 0 and self.y > 0
```

Figure 15 : Expression de la contrainte n 1 en OCL

- a. Si l'abscisse d'un point est inférieure à 100, alors son ordonnée doit l'être aussi

```
context Point inv :  
  self.x < 100 implies self.y < 100
```

Figure 16 : Expression de la contrainte n 2 en OCL

- b. Après appel de l'opération *moveto* avec les paramètres *a* et *b*, l'abscisse du point vaut *a* et son ordonnée vaut *b*.

```
context Point :: moveto(a : int, b : int)  
  post : self.x = a and self.y = b
```

Figure 17 : Expression de la contrainte n 3 en OCL

- c. L'opération *moveto* ne doit être invoquée qu'avec des paramètres positifs

```
context Point :: moveto(a : int, b : int)  
  pre : a > 0 and b > 0
```

Figure 18 : Expression de la contrainte n 4 en OCL

La notation pointée, qui est habituellement d'usage en programmation par objet, est déjà utilisée en Objective CAML pour les enregistrements et pour les modules.

- Object Modeling Technique, popularisée par Jim Rumbaugh.

- Object-Oriented Analysis and Design, fondée par Grady Booch.
- Object-Oriented Software Engineering, méthode créée par Ivar Jacobson.

V. CONCLUSION :

Comme **UML** n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente, en effet nous avons choisi le processus **UP** qu'on utilisera pour modéliser notre système. Cette modélisation sera détaillée dans le chapitre suivant.

Chapitre III : Modélisation du système

I. Introduction :

Dans ce chapitre nous introduisons l'analyse de notre système en utilisant Le **langage de modélisation unifié (UML)** « en s'appuyant sur le processus UP », est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet, en conséquence, nous allons détailler les trois étapes : tout d'abord, nous commencerons par définir le diagramme de cas d'utilisation, ensuite les cas d'utilisation vont être détaillés en plusieurs diagrammes de séquences, nous terminerons par représenter le diagramme de classe qui décrit la structure statique de notre système.

II. Environnement de modélisation :

Il existe plusieurs outils de modélisations **UML** qui sont disponibles comme logiciels payants ou gratuits, on peut citer : Frame UML, PowerAmc, UMLDesigner, ArgoUml, ModelMaker.

Modelmaker 11 : permet d'éditer les diagrammes d'**UML**, il inclut deux éditions : une édition pascal et une édition C# et de générer un code en pascal objet dédiée pour un environnement Delphi.

III. Modélisation de système :

Diagramme de cas d'utilisation :

Ce diagramme permet de décrire l'interaction entre les acteurs et le système.

➤ Identification des acteurs responsables matériels :

Un acteur est l'idéalisation d'un rôle joué par une personne ou un groupe de personnes. L'acteur qui interagit avec notre système est :

L'Administrateur : utilisateur de système.

➤ Identificateur des cas d'utilisateur :

Un cas d'utilisation présente une fonctionnalité du système, notre système comporte les cas d'utilisation suivant :

- Achat matériel et logiciel.
- Installation et désinstallation logiciel.
- Intervention informatique.

Les cas d'utilisation sont schématisés dans le diagramme représenté dans la figure 20 :

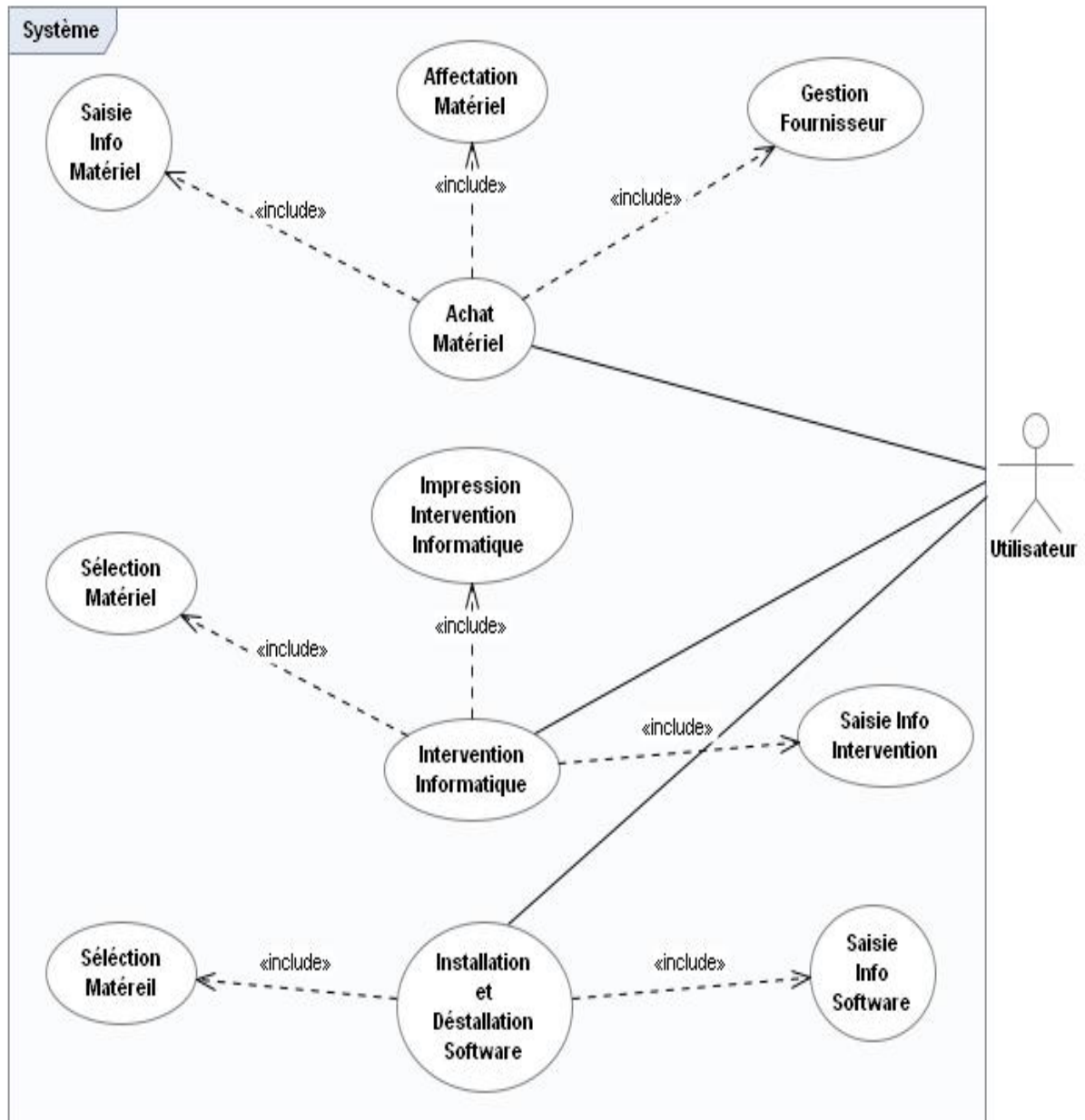


Figure 19 : Diagramme de cas d'utilisation

IV. Les diagrammes de séquence :

Dans cette partie nous allons présenter les interactions des objets du système par un diagramme de séquence pour chaque scénario de chaque cas d'utilisation.

1) Authentification :

Lorsque l'utilisateur demande l'accès à l'application, il doit tout d'abord saisir son Login et son mot de passe, s'il possède un compte, le système va afficher le menu principal de l'application sinon il affiche un message d'erreur. Ce scénario est présenté par le diagramme de séquence de la figure 21.

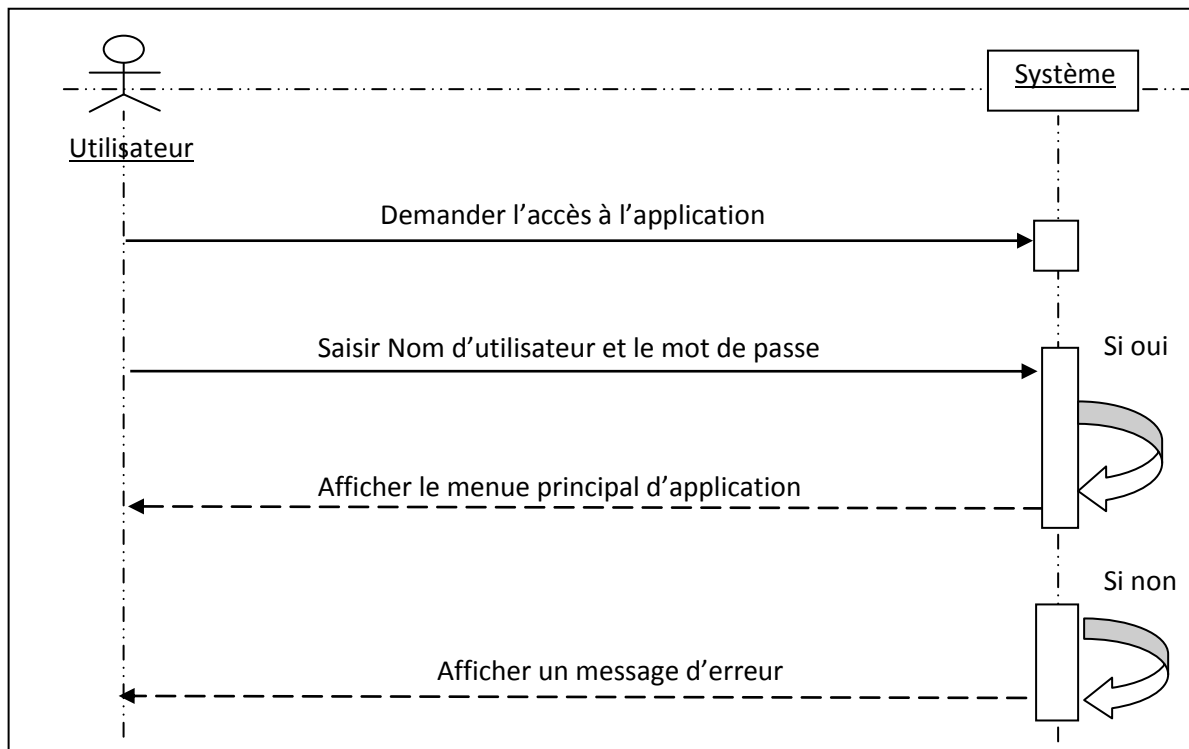


Figure 20 : Diagramme de séquence « authentification »

2) Affectation Matériel :

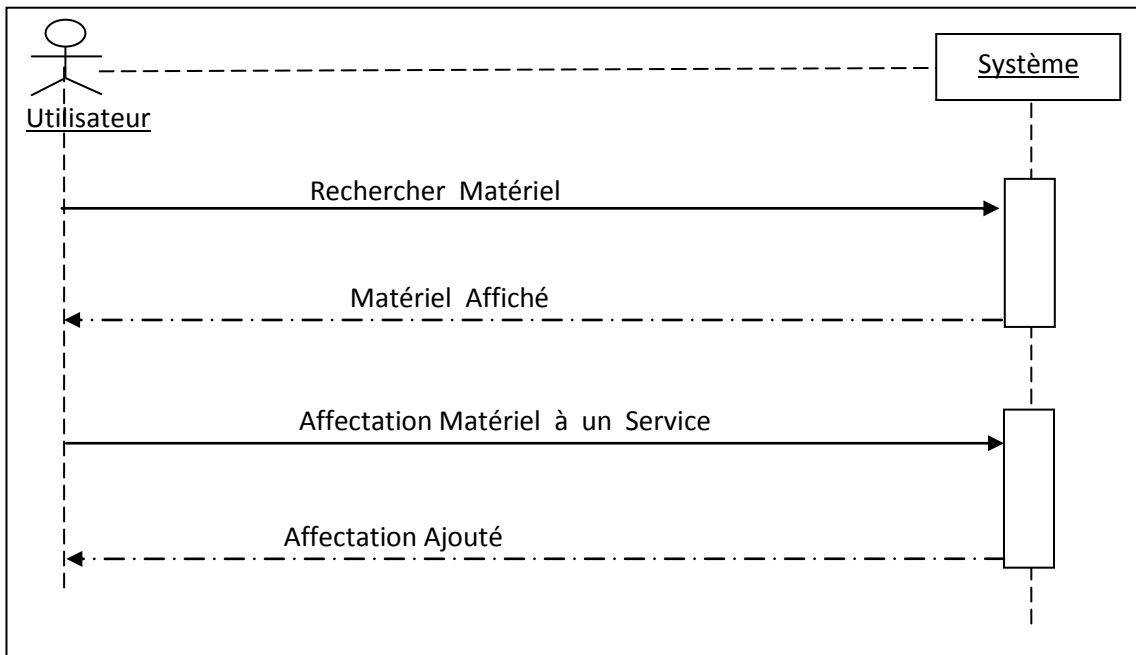


Figure 21 : Diagramme de séquence « Affectation matériel »

3) Installation _ Désinstallation :

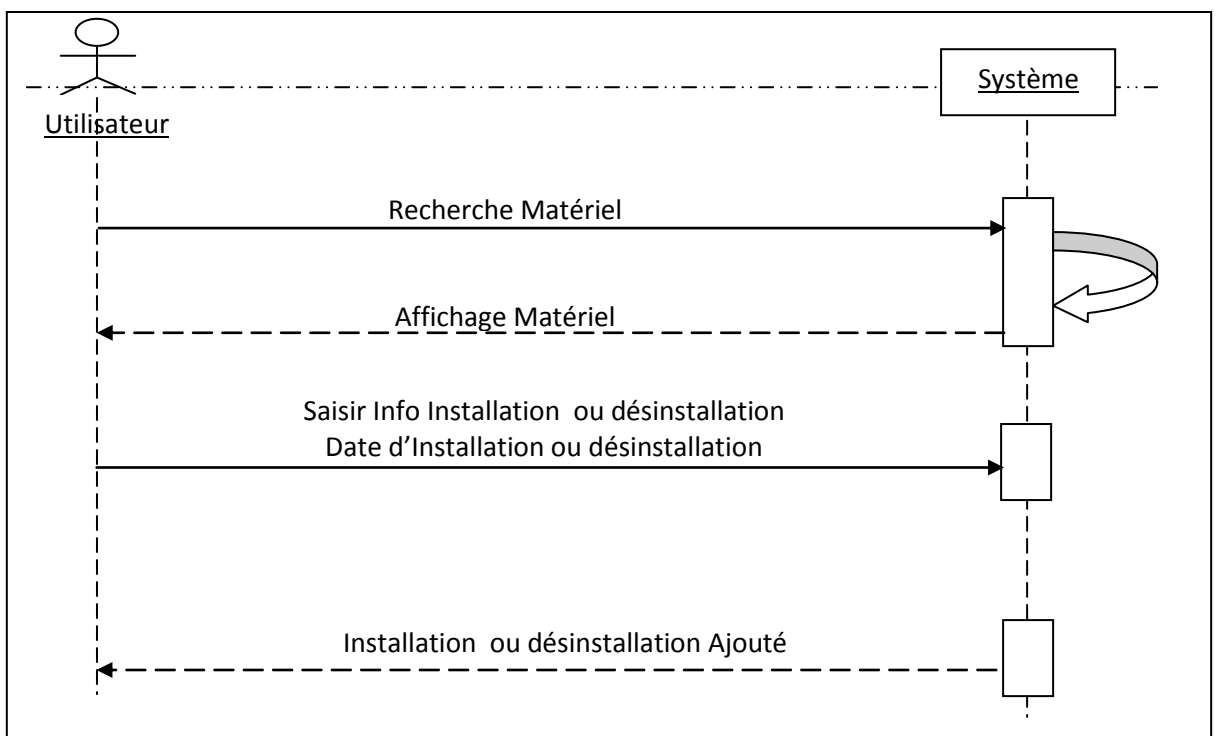


Figure 22 : Diagramme de séquence « Installation_Désinstallation »

4) Intervention :

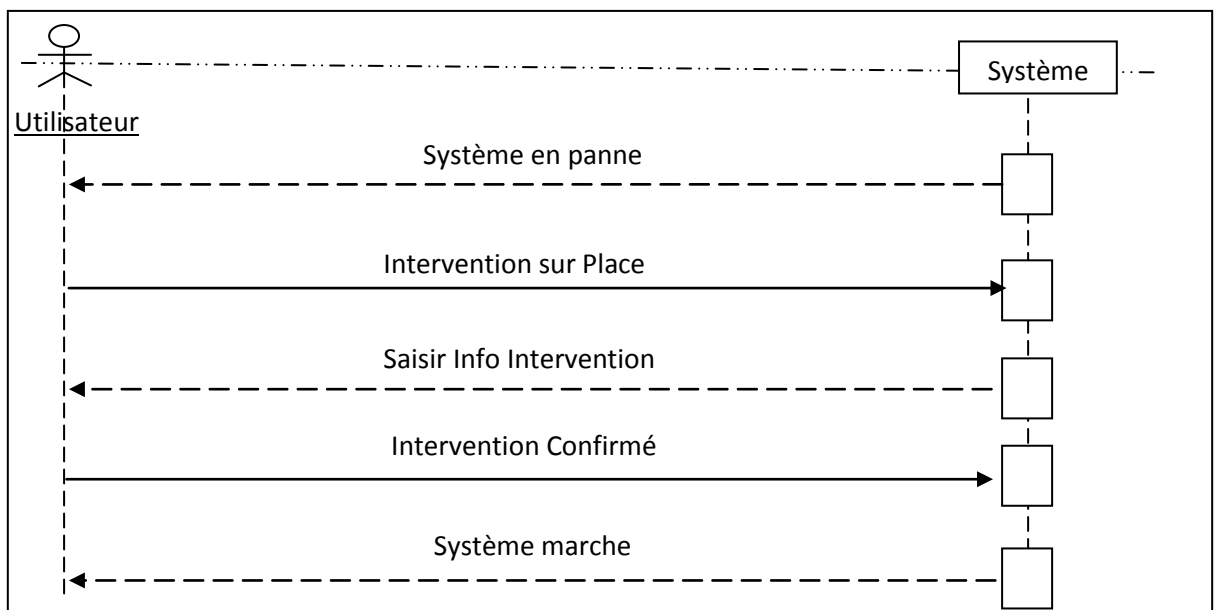


Figure 23 : Diagramme de séquence « Intervention »

V. Diagramme de Classe :

Le diagramme de classe identifié les classes de notre système et les associations entre elles.

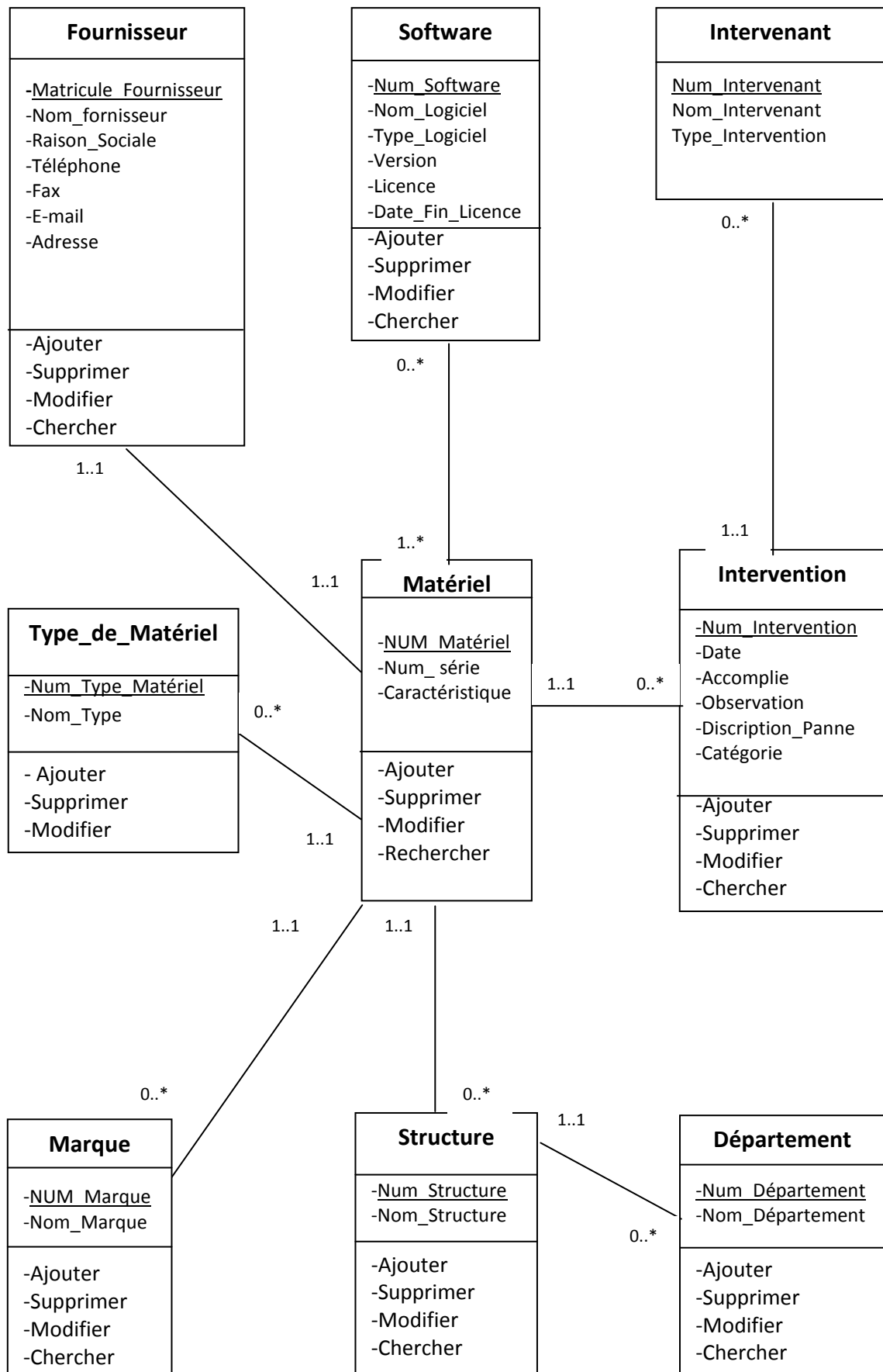


Figure 24 : Diagramme de Classe

VI. Modèle Logique des Données :

T_Matériels (Num_Matériel, Num_série, Caractéristique, NUM_Marque, Num_Structure, Num_Type_Matériel, Matricule_Fournisseur).

T_Fournisseur (Matricule_Fournisseur, Nom_fournisseur, Raison_Sociale, Téléphone, Fax, E-mail, Adresse).

T_Softwar (Num_Software, Nom_Logiciel, Type_Logiciel, Version, Licence, Date_Fin_Licence).

T_Marque (NUM_Matériel, Nom_Marque).

T_Structure (Num_Structure, Nom_Structure, Num_Département).

T_Département (Num_Département, Nom_Département).

T_Type_de_Matériel (Num_Type_Matériel, Nom_Type).

T_Intervention (Num_Intervention, Date, Accomplie, Observation, Discription_Panne, Catégorie, Num_Intervenant, Num_Matériel).

T_Intervenant (Num_Intervenant, Nom_Intervenant, Type_Intervention).

VII. Conclusion :

Dans ce chapitre nous avons présenté la modélisation de la structure statique et dynamique de notre système en utilisant un sous ensemble des diagrammes d'**UML** et en appliquant une partie du processus unifié comme démarche de conception et de modélisation. Cette conception est une étape nécessaire et très importante pour pouvoir créer notre base de données et réaliser l'application, le prochain chapitre détaillera les étapes de l'implémentation de notre application.

Chapitre IV : Implémentation du Système

I. Introduction :

Ce chapitre est consacré à l'implémentation de notre application qui s'appuie sur la modélisation présumé dans le chapitre précédant, pour l'implémentation nous avons utilisé Microsoft Visual Studio

II. L'environnement de programmation Microsoft Visual Studio 2010 sp1 :

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle **Visual Studio 2015**.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (**IDE**, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Développer [6].

III. L'environnement de gestion de base de donnée :

Nous avons utilisé le module de base de donnée **Microsoft SQL Server** est un système de gestion de base de données (abrégé en SGBD ou SGBDR pour « Système de gestion de base de données relationnelles ») développé et commercialisé par la société Microsoft [7].

Bien qu'il ait été initialement développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

Lors de sa création, Sybase SQL Server hérite des principes du moteur Ingres développé à l'origine par l'université de Berkeley.

IV. Description de l'application :

Nous allons présenter dans cette partie les principales fiches de l'application.

➤ Fiche d'authentification :

Cette fiche permet à utilisateur de s'authentifier pour pouvoir accéder aux autres interfaces du système :

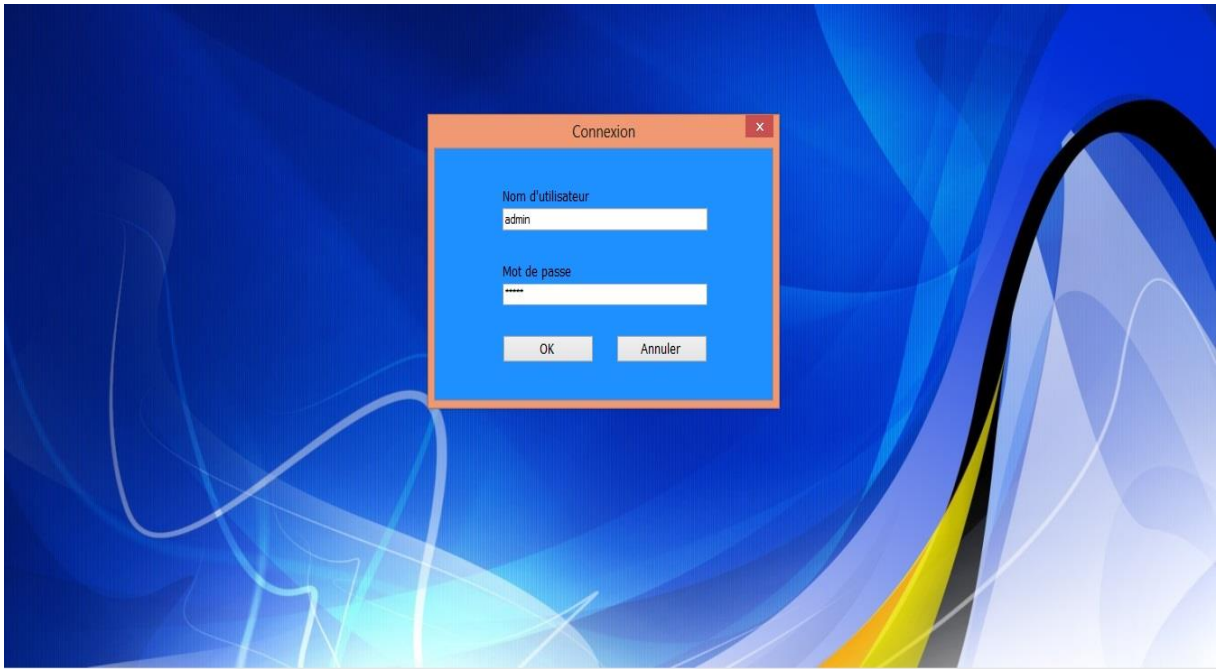


Figure 25 : Fiche d'authentification

➤ **Fiche principale**

C'est la fiche qui s'affiche lorsque l'utilisateur s'est authentifié correctement. A partir de cette fenêtre l'utilisateur peut accéder aux autres fiches de l'application en utilisant le menu

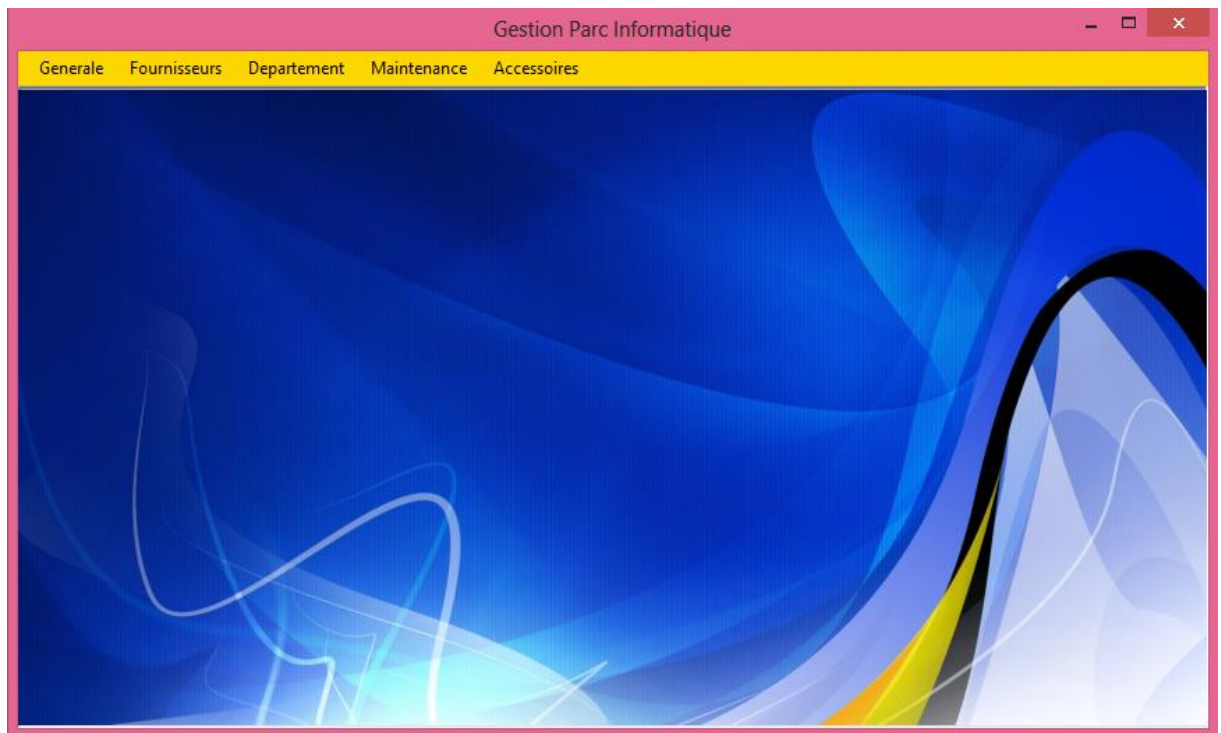


Figure 26 : Fiche principale

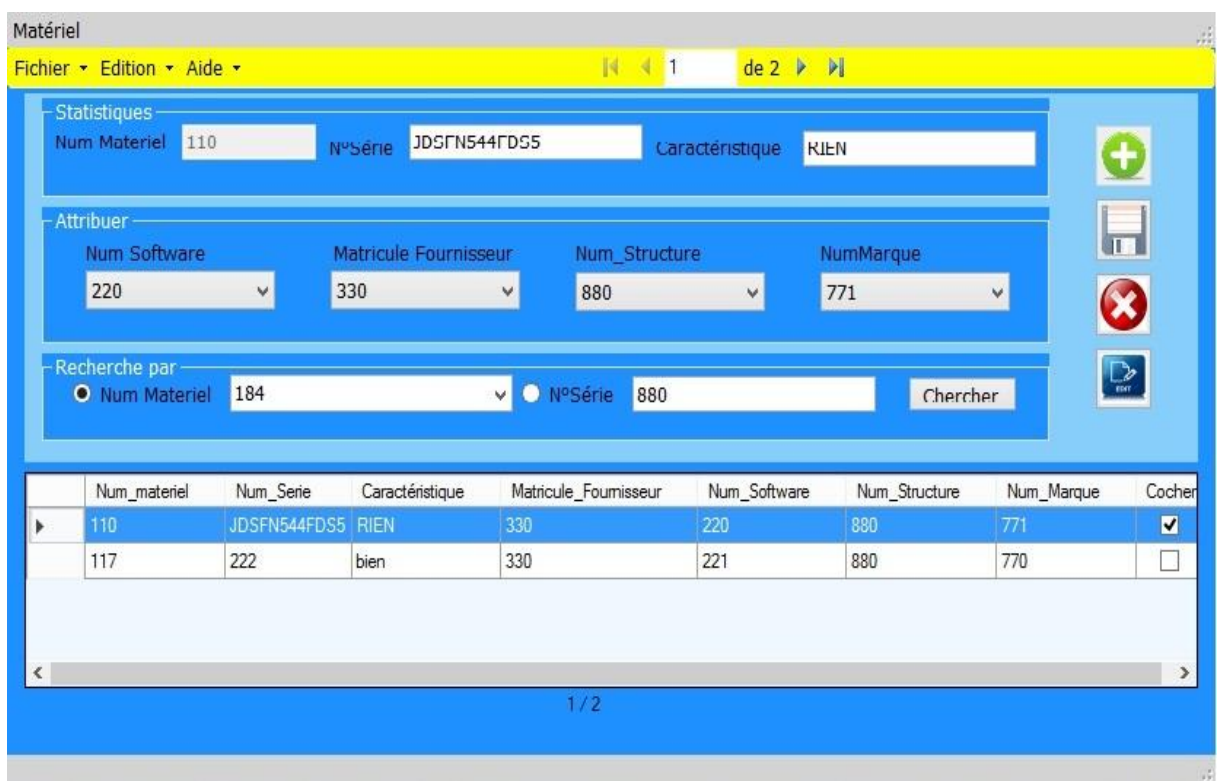


Figure 27 : Fiche Matériel

Fournisseur

Fichier ▾ Edition ▾ Aide ▾

Matricule_Fournisseur 330

Nom SolInf Tel 043208930

Raison_Sociale Tlemcen Fax 043205656

Adresse S3 Kiffane Tlemcen 13000

Email solinf@solinf.info

1 / 1

Matricule_Fournisseur	Nom_Fournisseur	Raison_Sociale	Adresse	Tel	Fax	Email	Cocher
330	SolInf	Tlemcen	S3 Kiffane Tlemcen 13000	0432089...	043205656	solinf@solinf.info	<input type="checkbox"/>

Figure 28 : Fiche Fournisseur

Intervention

Num_Intervention 440

Date 12/08/2015

Accomplie rien

Observation bien

Discription_Panne materiel

Catégorie 1546

1 / 1

Num_Intervention	Date	Accomplie	Observation	Discription_Panne	Catégorie	Cocher
440	12/08/2015	rien	bien	materiel	1546	<input type="checkbox"/>

Figure 29 : Fiche Intervention

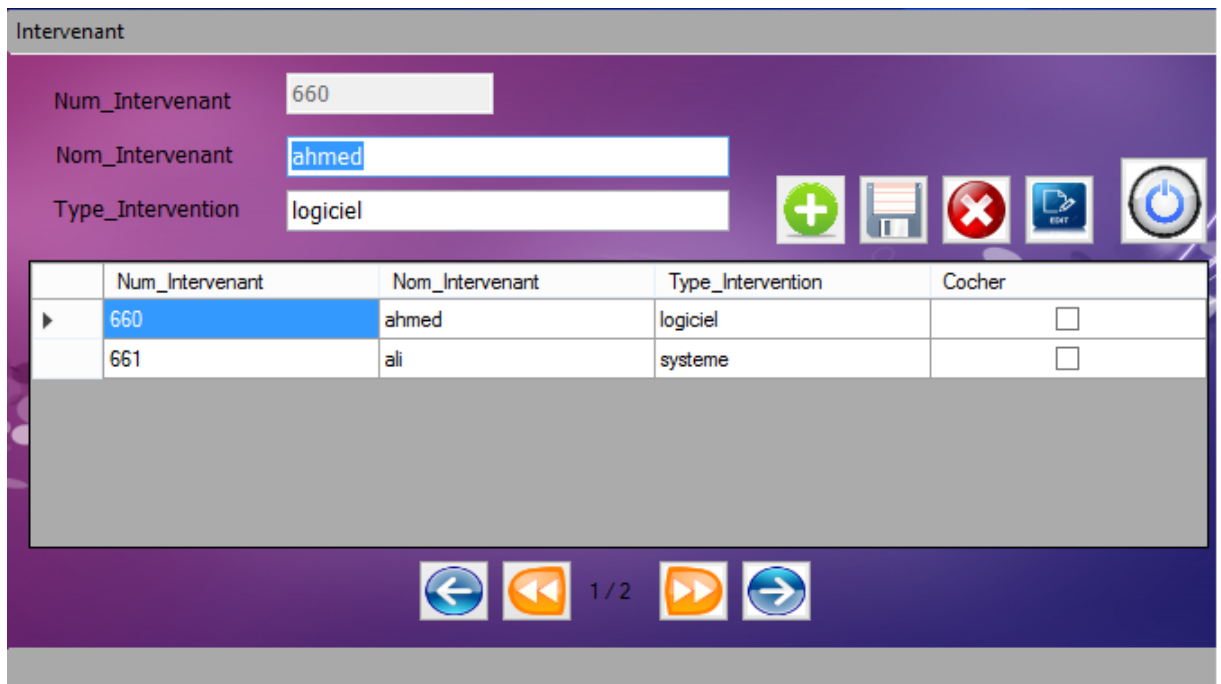


Figure 30 : Fiche Intervenant

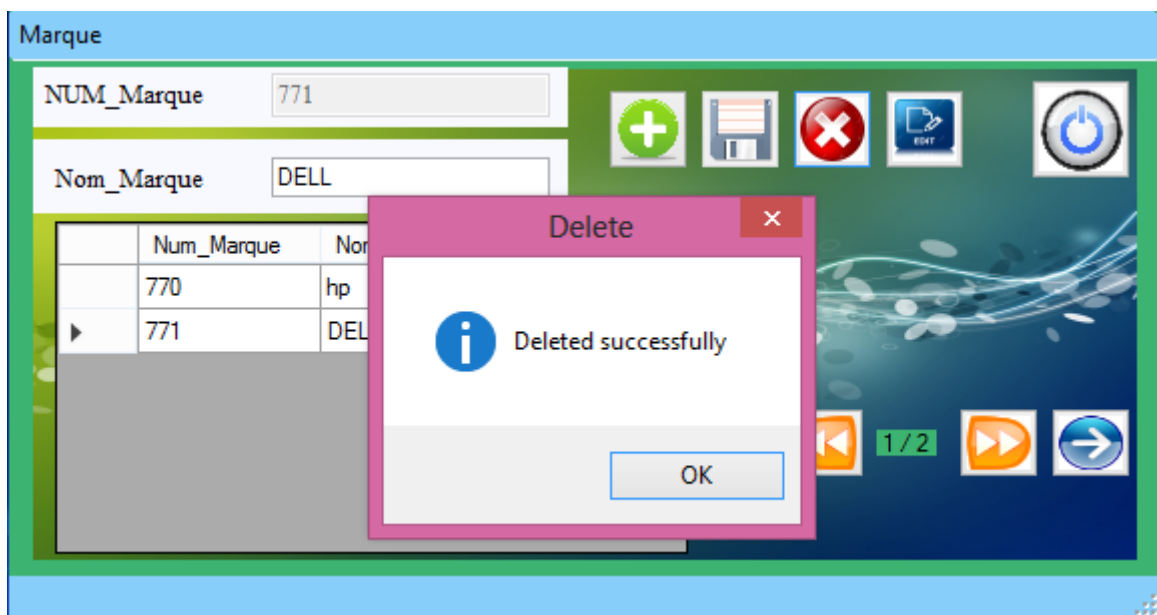


Figure 31 : Fiche Marque

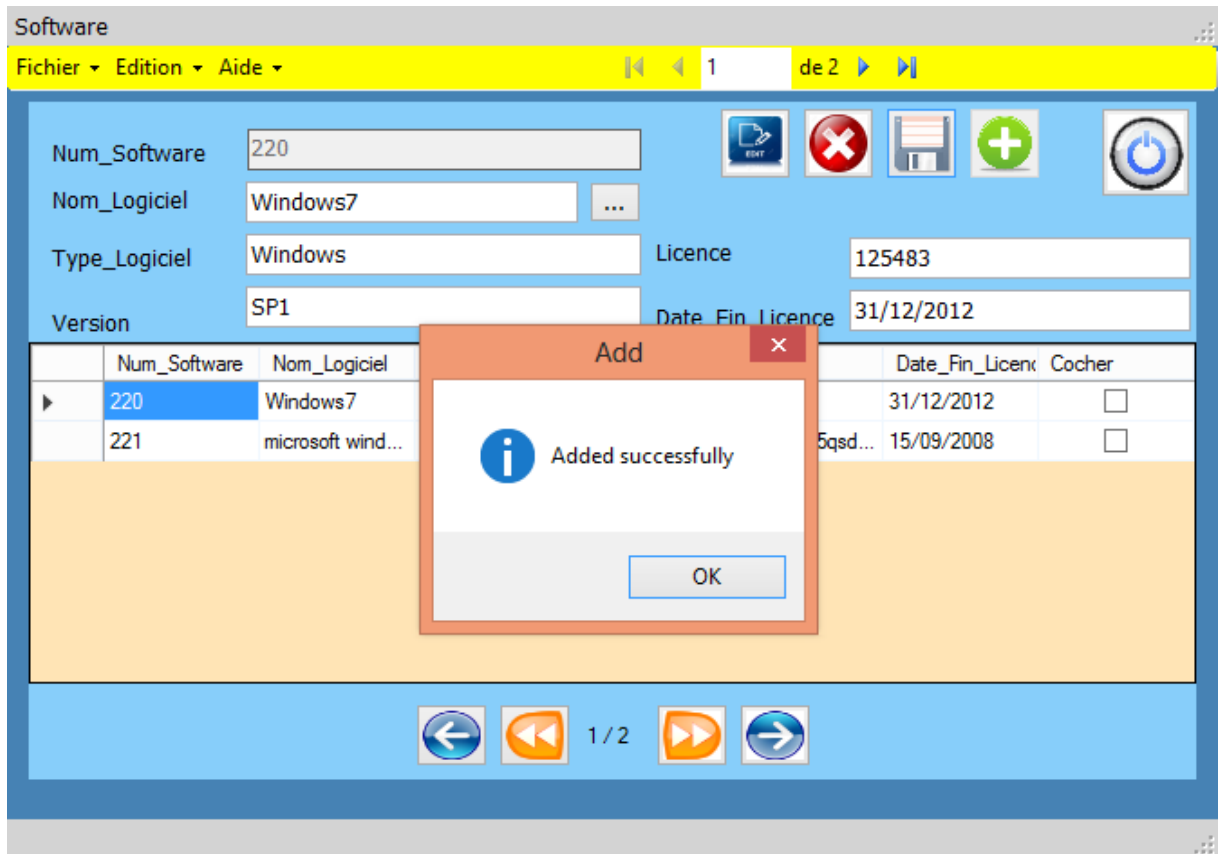
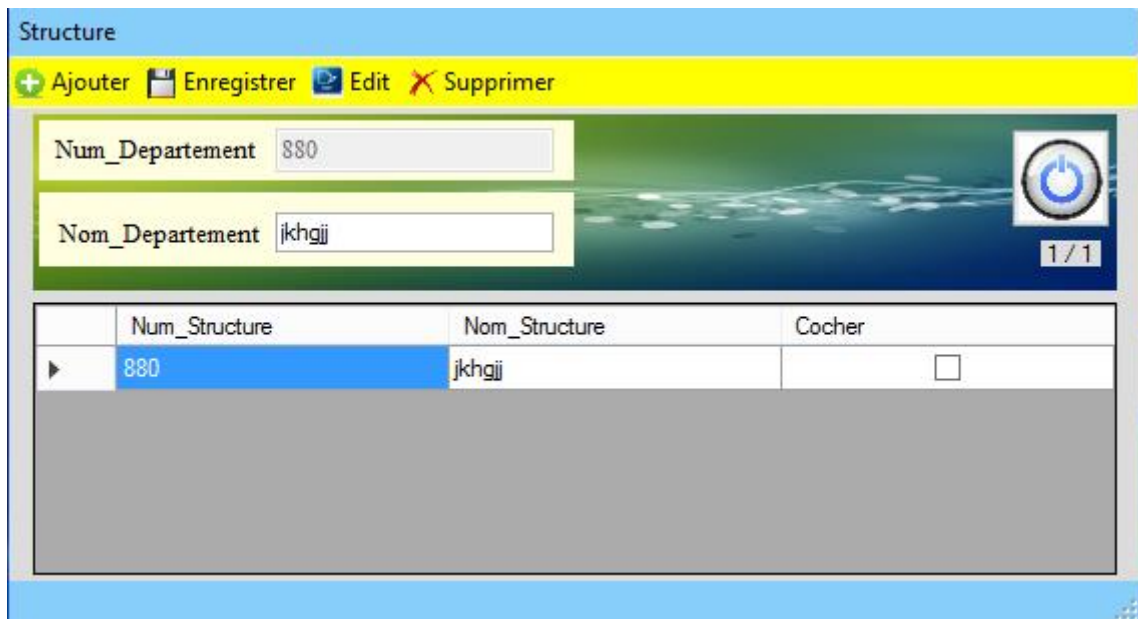


Figure 32 : Fiche Software



Figure 33 : Fiche Département



The screenshot shows a software window titled "Structure". At the top, there is a yellow toolbar with four icons and labels: "Ajouter" (Add), "Enregistrer" (Save), "Edit", and "Supprimer" (Delete). Below the toolbar, there are two input fields: "Num_Departement" with the value "880" and "Nom_Departement" with the value "jkhgij". To the right of these fields is a power button icon and a "1 / 1" indicator. Below the form is a table with the following data:

	Num_Structure	Nom_Structure	Cocher
▶	880	jkhgij	<input type="checkbox"/>

Figure 34 : Fiche Structure

V. Conclusion :

Dans ce dernier chapitre, nous avons présenté la partie réalisation de notre projet, et nous avons décrit les fiche les plus importantes de notre application.

Conclusion générale

Tout au long de la préparation de notre mémoire, nous avons essayé de mettre en pratique les connaissances que nous avons acquies durant nos études universitaires pour la gestion de parc informatique de l'Office de Promotion et de Gestion Immobilière.

L'analyse, la spécification, la conception sont faites orienter objet et décrites en UML en se basant sur le processus UP.

L'implémentation de l'application a été réalisée par l'environnement Microsoft Visual Studio, notre application a pour but principal de faciliter les tâches de l'administrateur de parc informatique (utilisateur de système). La base de données a été créée avec SQL server

Pendant la réalisation de notre travail nous avons acquis beaucoup de connaissances concernant le langage de modélisation UML et l'environnement Microsoft Visual Studio.

Nous avons souhaité d'avoir plus de temps pour mieux traiter le sujet proposé.

Mais nous espérons que notre travail sera évolué et amélioré par autres promotions et qu'il sera un aide pour eux.

Bibliographique

[1] Décret n°74-143 du 23 Octobre 1976 sur site : www.joradp.dz.

[2] Décret n°91-147 du Mai 1991 sur site : www.joradp.dz.

[3] Pièce-Alain Muller et Nathalie Gartner, «Modélisation avec UML», Presses de l'Université du Québec 2009.

[4] Yohann Richarde, Modélisation avec UML, Février 2000.

[5] Pascal Roques et Frank vallée, «UML en Action», Eyrolles 2002.

[6] Modelmaker 11.8.0 user Manuel Pascal and C# edition.

[7] Rudi Brucher, Optimiser SQL Server, Dunod.

Annexes