

Programmation en C-sharp d'un logiciel d'enregistrement des parcelles au ministère des affaires foncières dans la ville de Lubumbashi/Haut-Katanga

[Programming in C-sharp of a land registration software for the ministry of land affairs in the city of Lubumbashi/Haut-Katanga]

KASONGO NYANGE Franck¹ and SOMBODI BIN SELEMANI Chris²

¹Département des sciences Informatiques de gestion, ISC Lubumbashi, RD Congo

²Département des sciences commerciales et administratives et Informatique de gestion, ISP Lubumbashi, RD Congo

Copyright © 2023 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: The objective of this paper was to develop a computer application that would allow the various agents of the cadastral division in the city of Lubumbashi and the Director of Land Affairs to register information about the various people who come to apply for land, to facilitate the consultation of information, etc. The application would facilitate the registration of information about the various people who come to apply for land, to facilitate the consultation of information, etc. This application would facilitate the registration of information relating to the various people who come to apply for land, facilitate the consultation of information, etc. This solution is of crucial importance, as it will allow us to avoid the problem of land conflicts which has become commonplace in the city of Lubumbashi. It will also allow the registrar of real estate titles to verify the different plots of land that are available without having to move and in real time.

KEYWORDS: Programming, C-sharp, land registration, software, Lubumbashi, Haut-Katanga.

RESUME: L'objectif du présent article a consisté à mettre au point une application informatique qui permettra aux différents agents de la division de cadastre dans la ville de Lubumbashi et au Directeur des Affaires Foncières. Ladite application faciliterait l'enregistrement des informations relatives aux diverses personnes qui viennent effectuer de demande de terre, faciliter la consultation des informations, etc. Cette solution est une importance cruciale, car elle nous permettra d'éviter le problème de conflit parcellaire qui est devenu une monnaie courante dans la ville de Lubumbashi. Elle pourra permettre aussi au conservateur des titres immobiliers de faire de vérifications sur les différents lotissements qui sont disponible sans se déplacer et en temps réel.

MOTS-CLEFS: Programmation, C-sharp, cadastre, logiciel, Lubumbashi, Haut-Katanga.

1 INTRODUCTION

En Cette période du 21ème siècle, les domaines de la vie Informatique tend à devenir partout. En effet grâce à l'informatique chacun peut désormais disposer d'un outil performant pour la réalisation d'un très grand nombre des tâches avec efficacité et entre peu de temps.

C'est dans ce cadre que nous avons voulu aider les dirigeants de la division provinciale de titres immobiliers/ Lubumbashi en mettant un programme informatique qui aidera d'enregistrer les différents dossiers relatifs aux parcelles, les chercheurs et

les décideurs en mettant en permanence des informations fiables sur les concessions parcellaires. Ainsi le présent travail donne une approche informatique de l'étude de la faisabilité d'une telle application.

2 PROBLEMATIQUE

La problématique est définie comme un ensemble de préoccupations que se fait le chercheur par rapport à son thème, ou encore un genre des questions qu'il se pose en fonction des objectifs de recherche¹.

Dès nos jours, acquérir une parcelle en toute légitimité sur l'étendue de la province du haut-Katanga est un risque à prendre parce que les conflits fonciers inondent le secteur foncier et nous nous retrouvons dans un conflit où l'Etat a du mal à assurer l'appartenance d'une parcelle à une et une seule personne.

Cela s'explique par le fait que:

- La banque de donnée foncière est gérée manuellement les informations sont gardées sur papier dans des fardes;
- La consultation de toutes les données se fait manuellement, ce qui induit aussi un sérieux problème de mise à jour et de perte de temps;
- Les imprimés de valeurs foncières ne sont pas sécurisés et font l'objet de plusieurs falsifications. En vue de faire face à tous ces problèmes, nous allons réfléchir autour de la question suivante:

Quelle solution pourrions-nous mettre en place pour l'amélioration de la gestion des parcelles ?

3 MÉTHODOLOGIE

Pour toute recherche scientifique, quel que soit sa sphère d'application, son étendue ou son importance, il sied d'établir un protocole de conduite pouvant faire objet de la guidance de la thématique en étude jusqu'à l'obtention du rendu escompté. C'est la raison pour laquelle, pour ce qui notre thème, nous avons fait appel une méthode et une technique.

Pour ce qui concerne la méthode, nous avons choisi la méthode analytique, qui nous a permis par une démarche analytique de mettre en place une application informatique pouvant répondre à notre problématique et nous avons utilisé le Processus Unifié (UP) qui est un processus de développement logiciel utilisant UML (Unified Modeling Language) comme langage de modélisation facilitant ainsi la réalisation de cet article.

Pour ce qui concerne la technique, nous avons opté pour la technique de recherche documentaire, car c'est grâce à cette dernière que nous avons pu consulter un certain nombre des documents, notamment des ouvrages, des articles, des travaux de fins de cycle, des sites internet, dans le but de constituer une base documentaire pour la bonne élaboration de notre travail. Mais aussi la technique d'interview, qui nous permis d'entrer en contact avec certaines personnes victimes de conflit parcellaire.

4 PRESENTATION DE L'EXISTANT & CONSIDERATION THEORIQUES

L'étude préalable est appelée aussi l'étude d'opportunité est la première dans l'élaboration d'un projet informatique. Il est en effet indispensable de connaître l'organisation et son système traditionnel avant de prétendre à l'automatiser. C'est une étape, procédure et moyen utilisé pour réaliser les problèmes de la gestion que l'informatique aura envisagé. A l'issue de cette étude, une décision sera prise et elle portera sur les objectifs qui seront traité par l'application.

¹ KANYEMBO LWIMBA Castard, *Méthodes des Recherches Scientifiques*, Cours G2 IG, Institut Supérieur Pédagogique, Lubumbashi, 2010-2011, Inédit.

4.1 PRESENTATION DE LA DIVISION DE CADASTRE

4.1.1 HISTORIQUE

Dorénavant la division était dans la juridiction de la Circonscription foncière de Lubumbashi, plateau, pour des raisons d'ordre administratives et sociales pour rapprocher l'administration foncière aux administrés dans les domaines fonciers, il a plus à la hiérarchie de procéder à la subdivision de certaine circonscription foncière.

L'arrêté ministériel N° 020/CABMIN/AFF.FONC/2008 du 29 février 2008 portant création de la circonscription foncière de Lubumbashi/EST ville de Lubumbashi dans la province du Katanga portant sur la loi et les ordonnances ci-après:

- Vu la constitution de la RD Congo spécialement l'article 93
- Vu la loi N° 73/021 du juillet 1973 modifiée et complétée par la loi n° 80/008 du 18 juillet 1980,
- Vu l'ordonnance n°07/017 du 03 Mai 2007 telle que modifiée et complétée par l'ordonnance n°008/006 du 25 janvier 2008.
- Vu l'ordonnance n°007/018 du 16 Mai 2007 telle que complétée par l'ordonnance n°07/2007 du 25 janvier 2008.

4.1.2 SITUATION GÉOGRAPHIQUE

La division des titres immobiliers L'SHI/EST se situe au croisement des avenues KAPENDA et KAMBOVE dans la commune de Lubumbashi à Lubumbashi.

4.1.3 OBJECTIFS DE LA DIVISION

L'objectif de conservation des titres immobiliers est de s'occuper de la gestion du domaine foncier privé de l'Etat qui est constitué des terres urbaines et rurales que l'Etat a mis dans les commerces, c'est-à-dire susceptible d'être considérées à tout requérant (personne physique ou morale) privé ou public congolais. Ceci sous-entend que l'Etat est le seul propriétaire du fonds comme dit la loi foncière spécialement en son article 53. Les particuliers n'ont que le droit de jouissance et l'usage sur le sol de l'Etat et la priorité sur les immeubles incorporés au sol.

4.1.4 STATUT JURIDIQUE

La conservation des titres immobiliers est une entreprise publique

À caractère technique et social, dotée 'une personnalité juridique. Il est placé sous tutelle de l'Etat par l'entremise du ministère de mine et affaires foncières.

4.1.5 DOMAINE DE FONCTIONNEMENT

La division des titres immobiliers est dirigée par le chef de division appelé le conservateur des titres immobiliers, c'est lui qui représente la république dans la gestion de terre du domaine privé de l'Etat.

Le CTI est secondé par cinq chef de bureaux chargés respectivement de:

1. Services généraux personnel et comptabilité
2. Domaine foncier
3. Enregistrement et notariat
4. Contentieux fonciers et immobiliers
5. Taxation et recouvrement

Vu la pertinence des tâches qui lui sont reconnu et les attributions qui dévolues à ses compétences, il coordonne les activités de la division.

4.1.5.1 BUREAU SERVICES GÉNÉRAUX, PERSONNEL ET COMPTABILITÉS

Qui gère les personnels de la division et qui s'occupe de la comptabilité.

4.1.5.2 BUREAU DU DOMAINE FONCIER

Ce bureau s'occupe de la gestion du domaine foncier privé de L'Etat, et de l'étude des lotissements, l'examen et la facturation de tous les contrats d'occupation foncière, la tenue des registres fonciers.

4.1.5.3 BUREAU ENREGISTREMENT ET NOTARIAT

Ce bureau fait l'étude des aspects juridiques des titres fonciers et immobiliers.

4.1.5.4 BUREAU CONTENTIEUX FONCIER ET IMMOBILIERS

Il est chargé de traiter tous les conflits et litiges fonciers et immobiliers.

4.1.5.5 BUREAU DE TAXATION ET RECOUVREMENT

Ce bureau a comme attribution:

- Application de la police des tarifications;
- Suivi des paiements et recouvrement; - Perception des recettes de ka circonscription; - Perception des amendes transactionnelles.

5 ANALYSE DU METIER

5.1 VUE DESCRIPTIVE DU SYSTÈME

Le processus de demande de terre commence lorsque le CTI reçoit une demande de terre de la part d'un requérant, cette demande est instruite par le CTI qui la qualifie et transmet son accord ou son désaccord à l'intéresser.

En cas d'accord le CTI atteint la division du cadastre au travers une demande de travaux spécifiant clairement les travaux à exécuter. La division du cadastre envoie les techniciens sur terrains pour effectuer les travaux en rapport avec la demande des travaux, les travaux terminés, la division du cadastre transmet le dossier chez le CTI. Et le CTI vérifie les pièces conformément à sa demande de travaux.

Une fois que cette vérification s'avère fructueuse et que son avis soit favorable, le CTI au travers son bureau du domaine foncier prépare le projet de facturation pour permettre à son bénéficiaire d'être fixé sur les frais à payer (les frais connexes, le loyer anticipatif, les frais d'établissement contrat et les frais administratifs).

Le DOFO établit les contrats et le requérant paie pour le compte du trésor public et il signe sur le contrat de location ou d'occupation provisoire avec le CTI enregistre dans le registre des contrats.

5.2 CRITIQUE DE L'EXISTENT

Normalement les services du cadastre et des titres immobiliers travaillent en étroite collaboration, parce que sur le plan fonctionnel l'un ne peut rien sous l'autre quoi que c'est le service des titres immobiliers qui a la préséance sur le cadastre. Ce pendant nous remarquons qu'en pratique, le processus d'acquisition des parcelles n'est pas bien appliqué.

Parce qu'au-delà des dossiers que le conservateur des titres immobiliers peut ouvrir légalement suite à la demande de terre on a:

- Parfois les agents du cadastre vendent des parcelles sans signaler le conservateur des titres immobiliers.
- Parfois ce sont des chefs de quartier qui vendent des terrains aux particuliers sans informer le conservateur des titres immobiliers.

Au-delà de ces causes majeures des conflits fonciers, il faut citer aussi la plastification des imprimés de valeur mais non sécurisé et surtout la gestion de plan de lotissement est manuelle.

Ce qui fait qu'on peut avoir une même parcelle avec cinq numéros sur cinq copies différentes d'un même croquis. Selon la loi, n'est reconnu détenteur de la parcelle que celui qui est passé par le processus normale.

PROPOSITION DE SOLUTIONS

Notre apport a constitué en ce qui suit:

- L'application qui doit accompagner ce travail doit être en mesure de localiser sur une carte cadastrale le résultat d'une requête exécutée à une base de données parcellaire;
- Mettre en place un système d'information capable de faciliter la consultation les informations des données parcellaires.

6 CAPTURE DES DONNÉES DU SYSTÈME INFORMATIQUE

Cette étape consiste à:

La définition des besoins du système consiste à:

- Définir ce que le système doit faire
- Définir le contour du système
- Fournir la base de planification et le contenu technique de la définition des tâches qui seront informatisé par la sélection des certaines actions des modèles du métier.

Notre système informatique en construction sera à mesure d'automatiser les actions du modèle métier suivant:

- Rédiger demande de travaux;
- Etablir contrat;
- Transmettre dossier;
- Enregistrer contrat;
- Gérer utilisateur.

7 CONCEPTION DE SYSTEME INFORMATIQUE

Dans cette phase de conception, nous mettons au point la structure de la base de données.

7.1 DIAGRAMME DE CLASSE

Le diagramme de classe constitue l'un de pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable vis-à-vis des autres classes. Les traitements sont matérialisés par les opérations.

AJOUTER LES CLASSES

Une classe est une représentation abstraite d'un ensemble d'objets ayant les mêmes caractéristiques.²

Après notre analyse nous sommes arrivés aux entités suivantes:

REQUERANT DEMANDE DE TERRE

PARCELLE CONTRAT

DEMANDE DE TRAVAUX UTILISATEUR

² CHRISTIAN SOUTOU, UML2 pour les bases de données, Eyrolles, p.103.

AJOUTER LES ATTRIBUTS

Un attribut est une propriété élémentaire d'une classe, pour chaque objet d'une classe, l'attribut prend une valeur³.

Pour chaque classe nous avons comme attribut:

- **REQUERANT** (numéro, nom, post nom, prénom, téléphone, adresse, sexe, état civil, nationalité)
- **DEMANDE DE TRAVAUX** (numéro demande, date emission, date validité, motif)
- **PARCELLE** (numparcelle, nomlotissement, superficie, commune, adresse)
- **CONTRAT** (numcontrat, datecontrat, datefin,)
- **UTILISATEUR** (numutilisateur, nomutilisateur, adresse)
- **DEMANDE DE TRAVAUX** (numdemande, typedossier)

7.2 MODÈLE DU DOMAINE

Le modèle du domaine doit définir les classes qui modélisent les entités ou les concepts présents dans le domaine. Il s'agit donc de reproduire un modèle des objets du monde réel dans un domaine donné; l'élaboration du modèle de classe du domaine permet d'opérer une transition vers une véritable modélisation objet. Le modèle du domaine pour notre cas est le suivant:

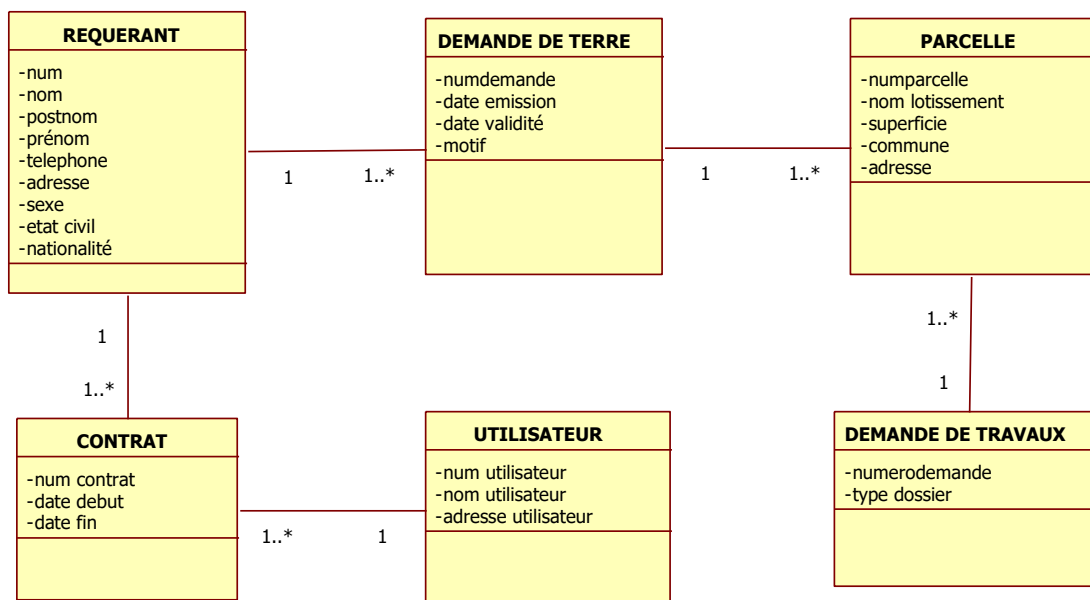


Fig. 1. Diagramme de classe modèle du domaine

7.2.1 LES CLASSES DE CONCEPTION

Les classes de conception sont apparentées au modèle du domaine mais en y ajoutant les opérations ou les méthodes.

³ XAVIER, Isabelle MOUNIER, UML2 pour les développeurs cours avec exercices corrigés, Eyrolles, Paris, 2007, p99

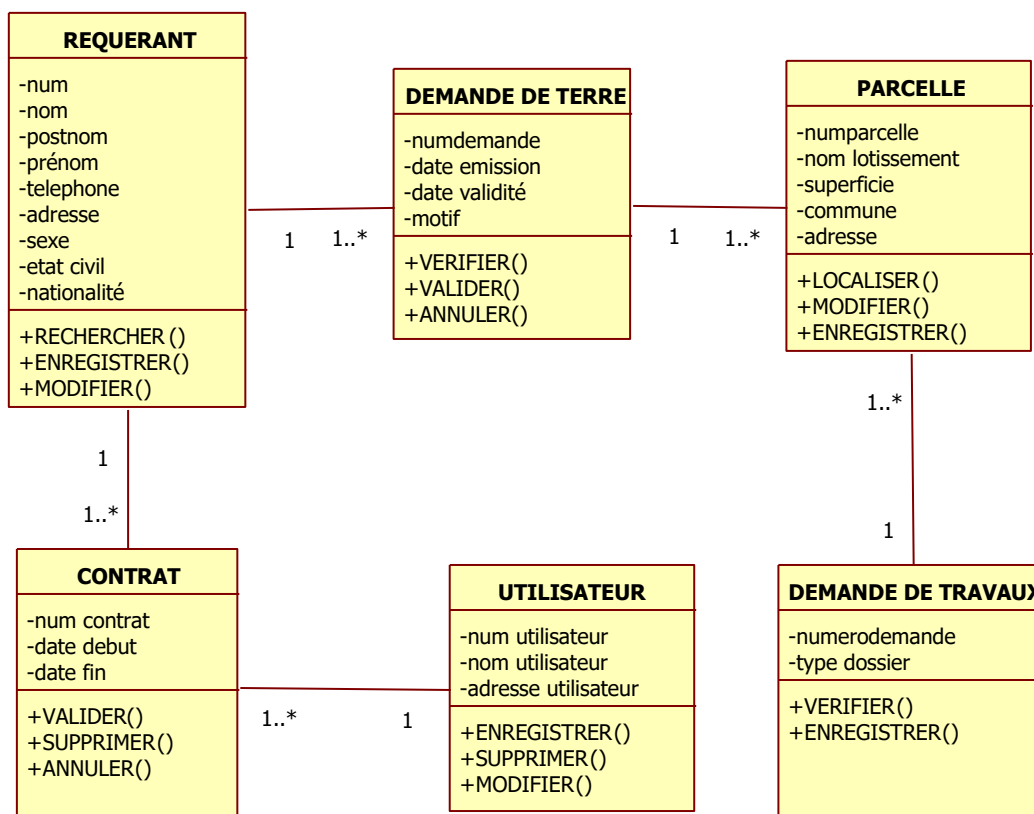


Fig. 2. Classe de conception

7.3 PASSAGE AU MODÈLE LOGIQUE DES DONNÉES

Dans cette partie, nous présentons les règles permettant de Décrire un modèle logique à partir d'un diagramme des classes UML. Il existe quatre règles pour traduire un schéma conceptuel UML

- En un schéma relationnel équivalent. Il est donc nécessaire du niveau conceptuel à un niveau plus porches des capacités informatiques. Ce niveau est appelé « Logique » et consiste de choisir l'un des modèles suivants:
- Modèle hiérarchique;
- Modèle réseaux;
- Modèle relationnel.

Chacun des modèles ci – haut reposent sur des techniques d'organisation particulières des données que des logiciels seront capables de gérer.

Pour notre travail, nous avons choisis l'approche des bases des données et précisément le modèle logique des données relationnel (MLDR) qui offre l'avantage de représenter facilement les données sous forme tabulaire et qui reste le modèle le plus adapté à la gestion⁴.

En plus, ce modèle a aussi un atout d'être pris en charge par la plus part des systèmes de gestion de base de données (SGBD) actuels.

⁴ GARDARIN G., *Bases des données : Objet et relationnel*, Eyrolles, Paris 1999, P 78

Pour ce faire, nous allons appliquer les règles de passage suivantes⁵:

a) Transformations des classes

Règle 1: Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant.

b) Transformations des associations

Règle 2: Dans une association un à plusieurs, il faut ajouter une clé étrangère dans la relations fils de l'association. Bref, la clé primaire de la relation père migre dans la relation fils.

Règle 3: dans une association plusieurs à plusieurs l'association devient une Relation dont la clé primaire est composée par la concaténation des identifiant des classes connectées à l'association.

Règle 4: en ce qui concerne l'association un à un, en vue d'éviter les valeurs nulles dans la base de données il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

En appliquant ces règles à notre modèle du domaine, nous obtenons le modèle logique suivant:

REQUERANT (num, nom, postnom, prenom, telephone, adresse, sexe, etat civil, nationalité)

DEMANDE DE TRAVAUX (numdemande, date emission, date validité, motif, #numrequerant)

PARCELLE (numparcelle, nomlotissement, superficie, commune, adresse, #numdemande, #numerodemande)

CONTRAT (numcontrat, datecontrat, datefin, #numrequerant, #numutilisateur)

UTILISATEUR (numutilisateur, nomutilisateur, adresse)

DEMANDE DE TRAVAUX (numdemande, typedossier)

8 REALISATION DE L'APPLICATION

Nous allons implémenter notre solution conçue au précédent dans un matériel et un logiciel précis. Il s'agit donc de définir à ce niveau l'architecture logicielle adoptée, l'environnement de développement de l'application choisi, ainsi que certaines caractéristiques techniques minimales du matériel informatique qui va abriter cette solution.

8.1 CHOIX DU LANGAGE DE PROGRAMMATION

Notre choix sur le langage de programmation a porté sur le C-SHARP noté par « C# ». En effet, C# est le langage par excellence de .Net, apparu en 2001. Le C Sharp est très polyvalent, Il permet de coder de simples applications consoles jusqu'à de gros programmes avec une multitude de fenêtres, en passant par les jeux.

Souvent, le langage C# a été comparé au java, pour qui, tout comme le C#, tout est objet. Cependant, le langage de .NET par ses améliorations au fur et à mesure de son existence, a su se forger son propre profil. En effet depuis sa parution, on est passé de la version 2.0 à 2.5, de 2.5 à 3.0 et enfin de la version 3.0 à 3.5, ayant dans chaque évolution bien entendu son nombre de nouveautés. De nouveaux mots clés sont ajoutés, de nouvelles sortes de classes et certaines syntaxes simplifiées... C'est donc au fil des années que ce langage devint de plus en plus efficace et plus facile d'accès que beaucoup d'autres langages orientés objet, ce qui fait aujourd'hui sa popularité.⁶

8.2 CHOIX DU SGBD

Pour notre application, nous avons choisi comme SGBD MySQL; MySQL est un système de gestion de bases de données (SGBD) qui gère pour nous les fichiers constituant une base, prend en charge les fonctionnalités de protection et de sécurité en fournit un ensemble d'interfaces de programmation, dont une avec PHP facilitant l'accès aux données⁷.

⁵ SOUTOU C, UML 2 pour les bases de données avec 2 exercices C, Ed. Eyrolles, 2010 P.128

⁶ www.Openclassroom.com

⁷ Joseph GABAY, UML2 Analyse et conception, Dunode, Paris, 104

L'expression SQL signifie Structured Query Language ou encore langage structuré des requêtes. MySQL consiste en un ensemble de programmes chargés de gérer une ou plusieurs bases de données, et qui fonctionnent selon une architecture client /serveur.

8.3 CHOIX DE L'ARCHITECTURE LOGICIELLE

Une architecture logicielle décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions.

Généralement une application informatique est subdivisée en 3 niveaux d'abstraction distincts ou couches qui sont la couche présentation, la logique application ou couche traitements et les données ou couche d'accès aux données.

- **La couche présentation**, appelée encore IHM⁸, permet l'interaction de l'application avec l'utilisateur. L'IHM doit toujours être ergonomique et conviviale. Cette couche gère les entrées venant du clavier, les actions de la souris et la présentation des informations à l'écran.
- **La logique applicative**, les traitements, décrivant les travaux à réaliser par l'application. Les traitements peuvent être découpés en deux familles:
 - ✓ Les traitements locaux, regroupant les contrôles effectués au niveau du dialogue avec l'IHM, visant essentiellement le contrôle et l'aide de la saisie;
 - ✓ Les traitements globaux, constituant l'application elle-même. Cette couche, appelée couche métier, contient les règles internes qui régissent une entreprise donnée.
 - ✓ Les données ou accès aux données, regroupant l'ensemble des mécanismes permettant la gestion des informations stockées par l'application.

AVANTAGES D'UNE APPLICATION EN COUCHES

En fait, il y a plusieurs avantages à utiliser cette technique:

- La maintenance des données est indépendante du support physique de stockage;
- La maintenance des traitements est simplifiée;
- La gestion des traitements depuis la couche de présentation est facilitée;
- Le travail en équipe est optimisé. C'est-à-dire lorsqu'un membre de l'équipe travaille par exemple sur la couche présentation, un autre peut travailler sur l'accès aux données et les autres travaillent sur les traitements sans perturber ce que les autres font;
- La migration d'un environnement graphique à un autre est relativement simple.

Ces trois niveaux d'abstraction peuvent se retrouver soit sur la même machine physique ou soit répartis entre plusieurs machines. C'est la manière de les découper ou de les répartir qui définit même le type d'architecture logicielle.

Selon la répartition de ces trois niveaux d'abstraction, nous distinguons:

- **Les architectures 1-tiers**
- **Les architectures 2-tiers**
- **Les architectures 3-tiers**
- **Et les architectures n-tiers.**

Pour ce qui nous concerne, nous avons opté pour l'architecture 2-tiers

Il s'agit d'une application client/serveur. Ce type d'architectures se compose de plusieurs clients et d'un serveur de données. Dans cette topologie, les clients se connectent au serveur à travers un réseau en utilisant des protocoles de réseau, tel que le TCP/IP.

⁸ IHM : Interface Homme Machine

Les clients implémentent l'interface utilisateur ou la couche présentation et contiennent la logique d'application (les traitements).

Le serveur lui, assure exclusivement la manipulation des données. On parle alors des clients lourds.

Exemple: Une application Win32 exploitant une base de données sur un serveur dédié.

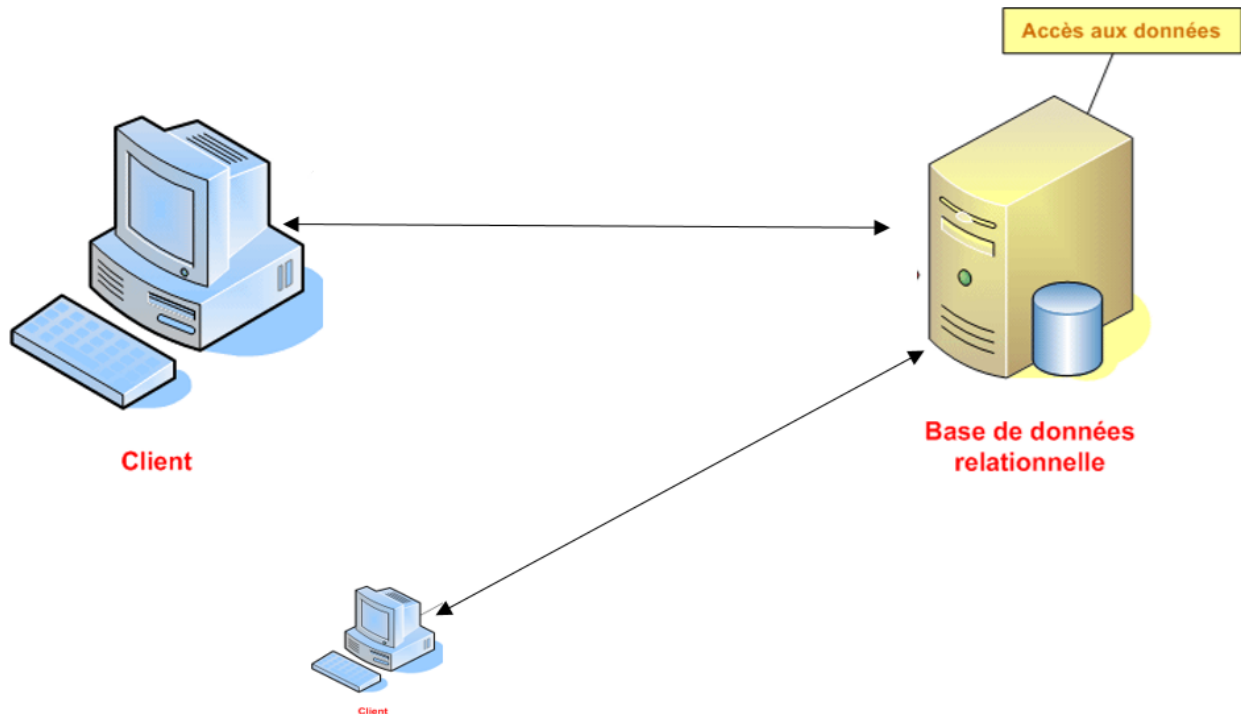


Fig. 3. Architecture 2-tiers

AVANTAGES

- Partage de données: toutes les données sont stockées à un seul endroit, au niveau du serveur, et sont accessibles à tous les utilisateurs.
- Passage à l'échelle: L'architecture à deux niveaux augmente le passage à l'échelle par rapport à l'architecture 1-Tier. L'architecture à deux niveaux peut s'adapter à environ 100 utilisateurs en permettant à plusieurs clients de se connecter au serveur.
- Mise à jour simplifiée: une modification apportée sur une information est suffisamment déployée dans tout le réseau, parce qu'elle a été effectuée au niveau du serveur.

INCONVÉNIENTS

- Le trafic élevé sur le réseau: du fait que des larges volumes de données sont transférés sur le réseau entre le client et le serveur, le débit élevé augmente la charge sur les ressources de réseau.
- La charge sur le serveur: suite à l'accès de tous les clients au serveur, la charge sur le serveur s'accroît et réduit son exécution.

8.4 CHOIX DU MATÉRIEL

A ce niveau nous proposons le minimum du matériel et du logiciel requis pour notre application, bref l'environnement dans lequel peut bien fonctionner l'application conçue à l'issue de notre travail. Nous pensons qu'un Pentium R peut déjà suffire à faire tourner l'application. Mais pour plus de performance, nous préconisons un matériel aux caractéristiques ci-après:

- Un microprocesseur **de la génération récente tel que i5 ou i7** dans le but de rendre fiable l'exploitation des informations;
- Un disque dur 20000 Giga-octets;
- Une mémoire RAM de 8 Giga-octets;
- Quatre machines correspondant au poste
- Un Access ne pointe « point d'accès »
- Une imprimante de marque EPSON Thermique et une autre pour l'impression des papiers A4
- Un système d'exploitation Windows 10 professionnel;

8.5 PRÉSENTATION DE QUELQUES INTERFACES DE L'APPLICATION

8.5.1 MODULE D'AUTHENTIFICATION AU SYSTÈME

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace GESTION_Parcelle
{
    public partial class Form_authen : Form
    {
        public Form_authen()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e){}
        public static string moncompte;
        public static string monpasse;
        public static string IP_SERVER;
        private void button1_Click(object sender, EventArgs e)
        {
            moncompte = txt_username.Text;
            monpasse = txt_pass.Text;
            IP_SERVER = textBox1.Text;

            try
            {
                if (verifier_champs() == false)
                {
                    if (conexion1.Maconnexion() == true)
                    {
                        SPLASH leon = new SPLASH();
                        bunifuTransition1.ShowSync(leon);
                        bunifuTransition2.ShowSync(leon);
                        leon.Show();
                        this.Hide();
                    }
                }
                else
                {
                    lbl1.Text = " le login ou le mot de passe est incorrect";
                }
            }
        }
    }
}
```

```

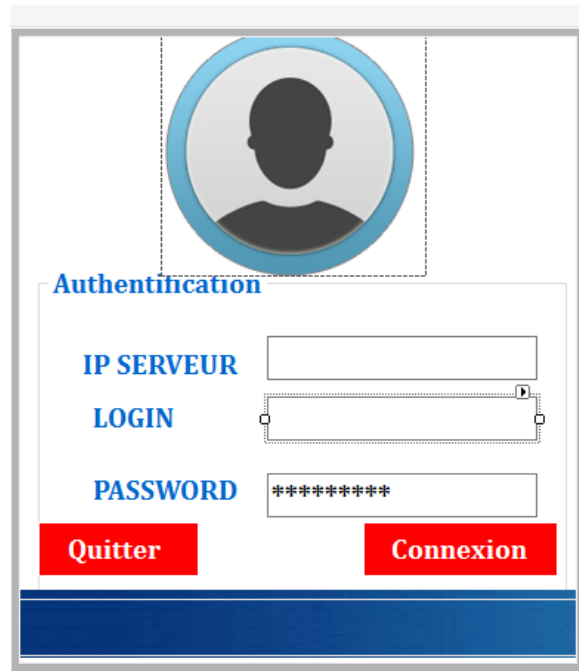
        effacer();
    }
}
else
{
    lbl1.Text = "veuillez remplir tous les champs";
    effacer();

}
}
catch (Exception michael)
{
    MessageBox.Show(michael.Message);
}
}
private void effacer()
{
    txt_username.Clear();
    txt_pass.Clear();
    txt_username.Focus();
}
private bool verifier_champs()
{
    bool retour = false;
    if (txt_username.Text == "" || txt_pass.Text == "")
    {
        retour = true;
    }
    else
    {
        retour = false;
    }
    return retour;
}

private void button2_Click(object sender, EventArgs e)
{
    {
        string message = "Souhaitez-vous vraiment Quitter cette application ?";
        string caption = "Attention";
        MessageBoxButtons buttons = MessageBoxButtons.YesNo;
        DialogResult resultat;
        resultat = MessageBox.Show(this, message, caption, buttons, MessageBoxIcon.Question);

        if (resultat == DialogResult.Yes)
        {
            Application.Exit();
        }
    }
}
}
}
}

```



8.5.2 CODE DE CONNEXION À LA BASE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
namespace impression
{
    class connexion
    {
        public static string marequette = null;
        public static MySqlConnection con;
        public static MySqlCommand cmd;
        public static MySqlDataReader Reader;
        public static MySqlDataAdapter dataadapter;
        public static string titreRapport;
        public static System.Data.DataSet DS;
        public static bool Maconnexion()
        {
            try
            {
                String constring =
                "server=localhost;database=cadaste;uid="+Form_authen.moncompte+"";password="+Form_authen.monpasse+"";
                con = new MySqlConnection(constring);
                con.Open();
                return true;
            }
            catch (Exception)
            {
                return false;
            }
            // Maconnexion() == true;
        }
    }
}
```

8.5.3 CODE POUR ENREGISTRER UN REQUÉRANT

```

public static class Gestion_parcelle
{
    public static void add_parcelle(string numreq,string nom, string postnom, string prenom, string tel, string adresse,
string sexe,string etatciv, string nationalite)
    {
        if (conexion.Maconnexion() == true)
        {
            try
            {
                string insert = "insert into requerant values('"+numreq+"','"+nom+"','"+postnom+"','"+prenom+"
','"+sexe+"','"+adresse+"','"+etatciv+"','"+nationalite+"')";
                connexion.cmd = new MySqlCommand(insert, connexion.con);
                connexion.cmd.ExecuteNonQuery();
            }
            catch (Exception jsein)
            {
                MessageBox.Show(m.Message, "CADASTRE ", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
    }
}

}public static void add_parcelle(string numreq,string nom, string postnom, string prenom, string tel, string
adresse, string sexe,string etatciv, string nationalite)
{
    if (conexion.Maconnexion() == true)
    {
        try
        {
            string insert = "insert into parcelle values('"+numreq+"','"+nom+"','"+postnom+"','"+prenom+"','"+sexe+"
','"+adresse+"','"+etatciv+"','"+nationalite+"')";
            connexion.cmd = new MySqlCommand(insert, connexion.con);
            connexion.cmd.ExecuteNonQuery();
        }
        catch (Exception jsein)
        {
            MessageBox.Show(m.Message, "CADASTRE ", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
    }
}
}

```

8.5.4 FORMULAIRE PRINCIPAL



FORMULAIRE DE DEMANDE DE TERRE

9 CONCLUSION

Partant de tout ce que nous venons de voir tout au long de notre travail intitulé « **Programmation en C-sharp d'un logiciel d'enregistrement des parcelles au ministère des affaires foncières dans la ville de Lubumbashi/Haut-Katanga** », on conclut comme suit:

La réalisation du présent travail est une succession d'étapes de modélisation ayant pour but l'obtention d'un produit fini répondant aux objectifs définis par la conservation des titres immobiliers Lubumbashi (EST) en matière de la distribution des parcelles.

Vu les difficultés que rencontre la Division des titres immobiliers dans la distribution des parcelles nous avons pris l'engagement de faire tout au long de notre travail, une application permettant de gérer les données relatives aux parcelles de la population

Pour ce faire, nous avons usage simple du langage UML qui nous a aidés à faire une petite conception partant de deux diagrammes.

Enfin, nous avons choisi des outils pour la réalisation du projet notamment:

- C#
- Mysql.

Nous ne prétendons pas avoir épuisé tous les problèmes ou failles auxquels fait face le service de CADASTRE dans sa gestion des conflits parcellaires et nous ne prétendons non plus avoir évolué sans embuches. Néanmoins, nous avons quand même fait l'idéale solution pouvant résoudre le problème relatif au conflit parcellaire dans un click grâce à notre application.

La balle est donc jetée à tout future chercheur qui voudrait bien aborder ce domaine de venir enrichir cette base que nous apportons et de corriger les failles qui s'y sont glissées sans notre volonté bien sûr, car seul le concepteur de l'univers est parfait.

REFERENCES

- [1] Blanc X., Mourier. UML2 pour les développeurs, éd, Paris, 2008, page 99.
- [2] Blanc X., Mourier I., UML 2 pour les développeurs, éd. Eyrolles, Paris, 2008, page 98.
- [3] CHRISTIAN SOUTOU, UML2 pour les bases de données, Eyrolles, p.103.
- [4] GARDARIN G., Bases des données: Objet et relationnel, Eyrolles, Paris 1999, P 78.
- [5] IVAR J., GRADY, B. et James, R., Le processus unifié de développement logiciel, Editions Eyrolles, 1999.
- [6] Joseph GABAY, UML2 Analyse et conception, Dunode, Paris, 104.
- [7] Roques P., UML 2: modéliser un application web, éd. Eyrolles, Paris, 2008.
- [8] Roques P; UML2: modélisé une application web -, éd, Eyrolles, Paris, 2008, page 5.
- [9] SOUTOU C, UML 2 pour les bases de données avec 2 exercices C, Ed. Eyrolles, 2010 P.128
- [10] Pinto et GRAWITE, Méthodes de recherches en sciences sociales, Dollaz, Paris 1971, page 289.
- [11] XAVIER, Isabelle MOUNIER, UML2 pour les développeurs cours avec exercices corrigés, Eyrolles, Paris, 2007, p99.
- [12] Franck KASONGO, *cours de programmation C#*, G3 IG/ISC 2020 – 2021, inedit.
- [13] KANYEMBO LWIMBA Castard, *Méthodes des Recherches Scientifiques*, Cours G2 IG, Institut Supérieur Pédagogique, Lubumbashi, 2010-2011, Inédit.
- [14] KASONGA, P., *Questions spéciales de conception de Système Informatique*, Cours de L2 I.G, Institut Supérieur Pédagogique, Lubumbashi, 2013-2014, Inédit.
- [15] http://www.notre-planete.info/terre/outils/sig_1.php.
- [16] <http://www.notre-planete.info/terre>.
- [17] <http://www.lecompagnon.info>.
- [18] <http://UML.free.fr>.