

# IMPLEMENTATION D'UN SYSTEME CLIENT-SERVEUR A DEUX NIVEAUX DE GESTION INTEGREE DES PRESENCES DES ETUDIANTS AU SEIN DE L'INSTITUT SUPERIEUR DE COMMERCE DE KISANGANI

**Mutongwa Kambale Dieu Merci<sup>1\*</sup>, Esabe Mandebo Rodriguez<sup>1</sup>**

*\*Sciences Informatiques, Département de l'Informatique de Gestion, Conception des Systèmes d'Information, Institut Supérieur de Commerce de Kisangani en DR Congo.*

**\*Corresponding Author:**

---

## **Résumé**

*Au début de chaque année l'appariteur doit procéder à l'enregistrement des tous les enseignants qui vont dispenser cours, mais aussi il doit enregistrer tous les étudiants régulièrement inscrits. Avec comme but de bien pouvoir gérer les présences des étudiants pendant ou après la prestation d'un cours.*

*Jusqu'à l'heure où nous parlons, cette gestion de présences des étudiants reste encore manuelle, et cela pose beaucoup de difficultés : Les listes des présences ne sont pas produites automatiquement, toutes les listes sont remplies manuellement, avec beaucoup de surcharges, et beaucoup de risque de redondances de noms. Chaque fois on doit élaborer les listes des présences à l'intention des étudiants. Comme ces listes des présences sont élaborées et remplies manuellement, elles subissent des mouvements à tout moment et ne porte quand, elles peuvent être victimes de perte et d'autres accidents possibles. Et une fois que les liste des présences sont perdues ce sont les étudiants qui sont victimes de plusieurs problèmes notamment : l'affichage injuste de  $\frac{3}{4}$  par l'enseignant titulaire du cours. Nous avons remarqué que, quand il y a la perte d'une liste, l'enseignant ne sait plus distinguer ceux qui étaient au cours et ceux qui n'y étaient pas, et voilà l'occasion pour lui de se faire l'argent en sélectionnant au hasard les noms auxquels il aurait affiché dans la liste de  $\frac{3}{4}$ . Ce qui reste hors de la vérité. Nous avons remarqué aussi qu'il n'y a pas un bon archivage auquel on airait fait recourt au cas de perte des listes dû au mauvais système actuel de conservation des listes de présences qui demeure archaïque.*

## INTRODUCTION

Afin de vouloir gagner le temps, de conserver les données, de limiter le nombre d'employés et pas mal d'autres raisons, les petites, moyennes et grandes entreprises sont bien poussées à chercher des solutions informatiques capables de répondre à leurs besoins. Cela dit, les entités congolaises doivent prendre conscience des avantages qu'elles gagneraient d'une gestion automatisée des présences et la considérer avec plus d'estime. C'est dans ce cadre que s'inscrit notre projet de recherche qui consiste à doter l'institut supérieur de commerce de Kisangani d'une application informatique de gestion des présences afin de résoudre quelques problèmes qui ne cessent de survenir.

Cette expansion considérable résulte du fait que ses engins sont aptes à traiter toute information de manière très rapide et rationnelle. Ainsi, savoir gérer l'information est l'une des activités qui, actuellement procure le développement rapide aux institutions et entreprises. C'est dans cette perspective que les hommes ont jugé bon de traiter automatiquement les informations pour en tirer le maximum de profit sans beaucoup de manipulations fastidieuses.

### 0.1. Problématique

Il n'existe pas de système automatisé de gestion de présences au sein de l'Institut Supérieur de Commerce de Kisangani. Toutes les informations concernant ce sous domaine sont stockées manuellement et moins sécurisés.

Après toutes nos observations effectuées, nous avons constaté l'absence total de système automatisé des présences au sein de l'institution ce qui engendre un certain nombre de problèmes tels que :

- Elaboration de liste des présences manuelles qui occasionne des erreurs ;
- La lenteur dans l'accès aux données ;
- Le risque de perte d'informations ; - Un archivage moins sécurisé.

Eu égard à ceux qui précèdent et à l'importance de la problématique dans tout travail scientifique et compte tenu de nos observations faites au sein de l'Institut Supérieur de Commerce de Kisangani, nos questions s'articulent autour de préoccupations suivantes :

- Comment faire pour booster la rapidité et diminuer efficacement la lourdeur d'obtenir une liste des présences ?
- Quel type de solution pouvons-nous mettre en place pour bien conserver les informations liées aux présences des étudiants et permettre un accès multiple aux données des étudiants via différents points d'accès ?

### 0.3. Hypothèses

Pour répondre de manière provisoire aux questions de notre problématique, nous proposons ce qui suit :

- Mettre en place une application informatique conçue en WLangage, connectée avec HFSQL-Classic0 (Hyper File SQL) considéré aussi comme SGBD serait une solution escomptée pour produire facilement et rapidement les listes des présences ;
- Pour permettre un accès multiple aux données relatives aux étudiants et garantir la sécurité des données, il serait préférable de mettre en place une application du type client-serveur à deux tiers ;

### 0.4. Méthodes utilisées

Il existe plusieurs méthodes, mais au cours de cette recherche, nous avons utilisé les méthodes suivantes :

- ❖ **Méthode historique** : nous a aidé à connaître l'évolution de cette institution dans le temps et dans l'espace, afin de mieux comprendre sa situation actuelle et son emplacement géographique.
- ❖ **Méthode structurale fonctionnelle** : nous a permis de connaître avec facilité la structure organisationnelle de l'ISC Kisangani ainsi que la répartition des tâches et de l'attribution de chaque structure.
- ❖ **Méthode UP** : nous a permis de planifier le projet en proposant de démarche rationnelle avec identification d'un chemin critique.

### 0.5. Objectifs poursuivis

Toute dissertation scientifique poursuit des objectifs qui constituent sa ligne directrice. Cette étude vise à faire une application qui aidera l'Institut Supérieur de Commerce de Kisangani à bien gérer les informations des présences des étudiants.

Dans le cadre de cette recherche, notre objectif est de :

- Muter cette gestion manuelle de présences des étudiants vers un système automatisé ;
- Concevoir et réaliser une application intégrée pour la gestion des présences des étudiants au sein de l'institut supérieur de commerce ;
- Déployer un système interactif distant des opérations des présences.

## I. CONCEPTION DU SYSTEME D'INFORMATION

Nous allons analyser et concevoir *JeSuisPrésent* avec les diagrammes UML (diagrammes statique et dynamique) en suivant la méthode de processus unifié qui est processus de développement logiciel construit sur UML ; il est itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques.

UML est un langage de modélisation graphique. Il est apparu dans le monde du génie logiciel, dans le cadre de la conception orientée objet. Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes. En effet, l'UML permet une meilleure conception du côté de l'application avec ses notions d'objets et de classes, et nous donne une décomposition claire et simple afin de dégager les entités et les classes nécessaires.

**I.2. MODELISATION DYNAMIQUE**

**I.2.1. Diagramme de cas d'utilisation**

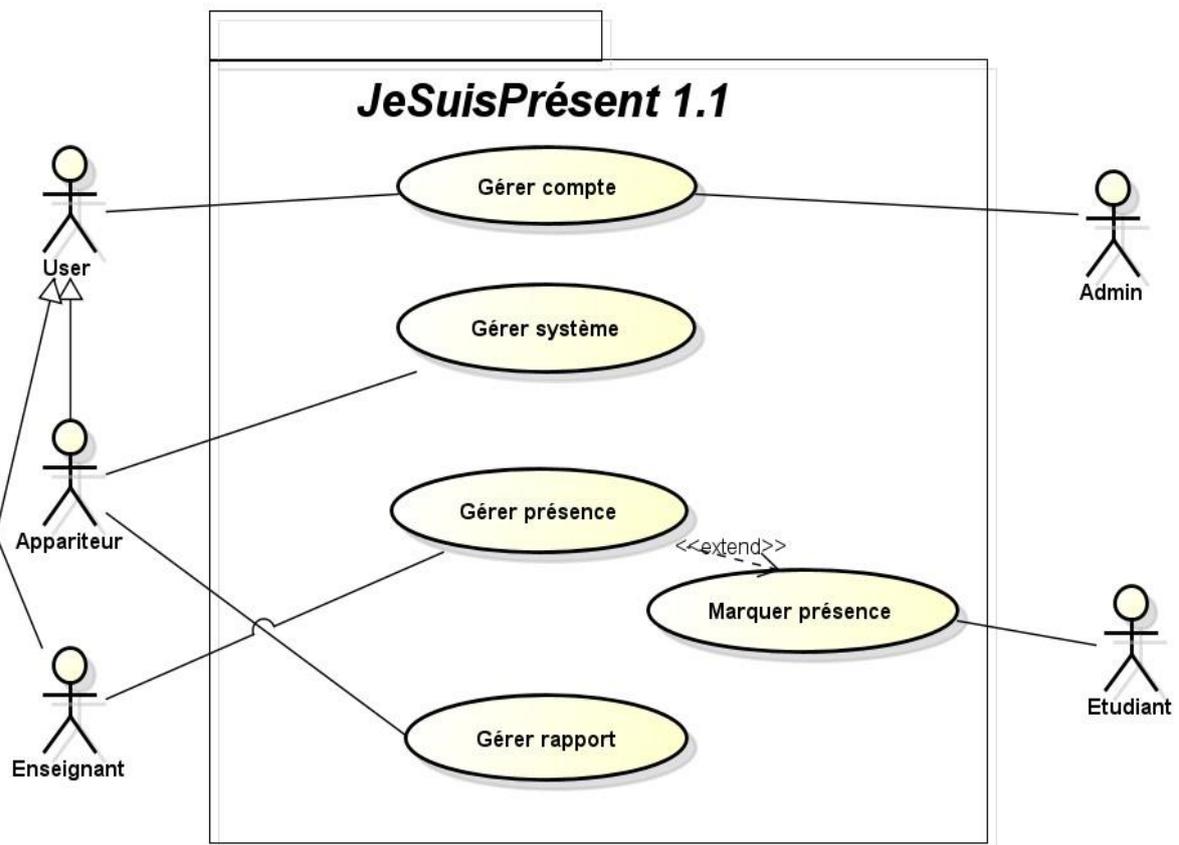
Les **diagrammes de cas d'utilisation** (DCU) sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

Numéro	Cas d'utilisation	Acteurs
1	Gérer compte	Utilisateur/Admin
2	Gérer système	Appariteur
3	Gérer présence	Enseignant- Etudiant
4	Gérer rapport	Appariteur

**Tableau 1: Recensement des acteur et leur rôle**

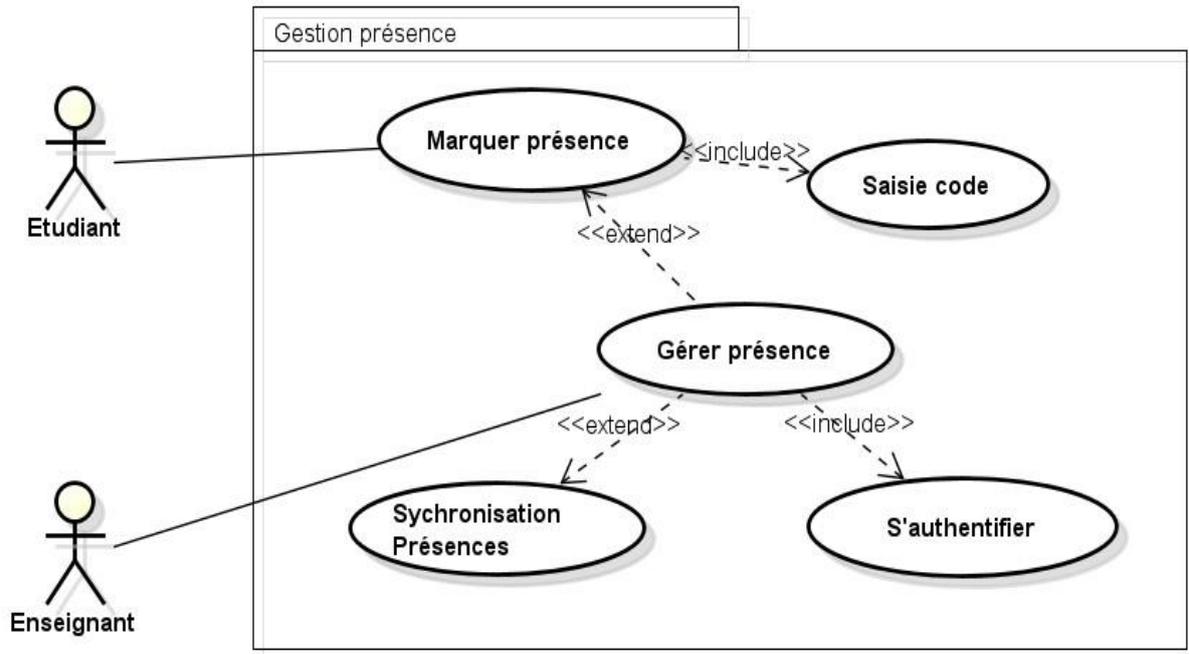
Après identification des acteurs et des cas d'utilisations, nous élaborons le digramme des cas d'utilisation globale puis nous parcourons les itérations par paquet. Chaque paquet aura quelques descriptions textuelles et leurs descriptions graphiques seront faites dans les diagrammes suivants. Signalons aussi que ces modèles ont été tracés à l'aide du logiciel AstahUML.

□ **Diagramme de cas d'utilisation global**



**Figure 1 : Diagramme de cas d'utilisation global**

□ **Itération : Gestion présence**



**Figure 2 : Itération gestion presence**

➤ **Cas d'utilisation : Marquer présence**

**Acteurs :** Enseignant et Etudiant

**Objectif :** Permettre à l'étudiant de signer sa présence après chaque prestation de cours enseigné par l'enseignant.

**Précondition**

- Le système doit être disponible - Prestation cours
- Authentification de l'enseignant

**Post-condition**

- Enseignant authentifié
- L'interface de présence s'affiche

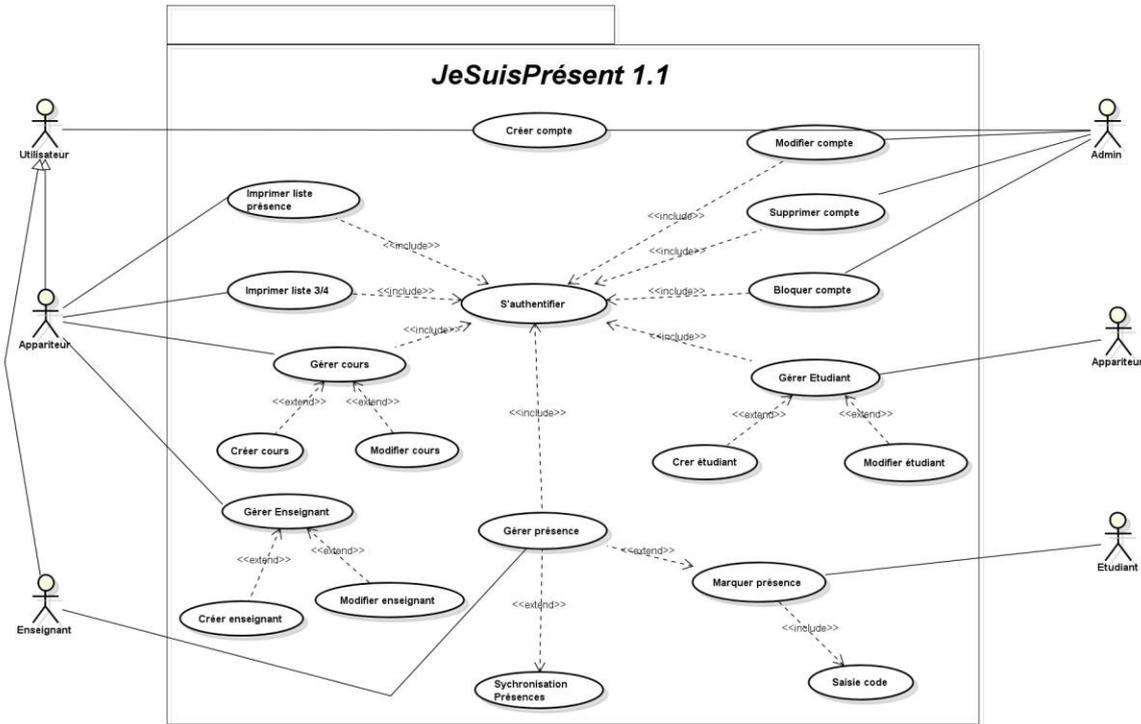
**Scénario nominal**

1. Demande formulaire
2. Système affiche le formulaire présence
3. Saisie code étudiant
4. Valider
5. Système vérifie le code étudiant
6. Le système affiche le message succès

**Scénario alternatif**

- Scénario Alternatif SA1 : Code erroné
- SA1 : commence au point 5 du scénario nominal
- Le système affiche le message d'erreur
- Le scénario alternatif reprend au point 3 du scénario nominal

➤ **Diagramme de cas d'utilisation détaillée**



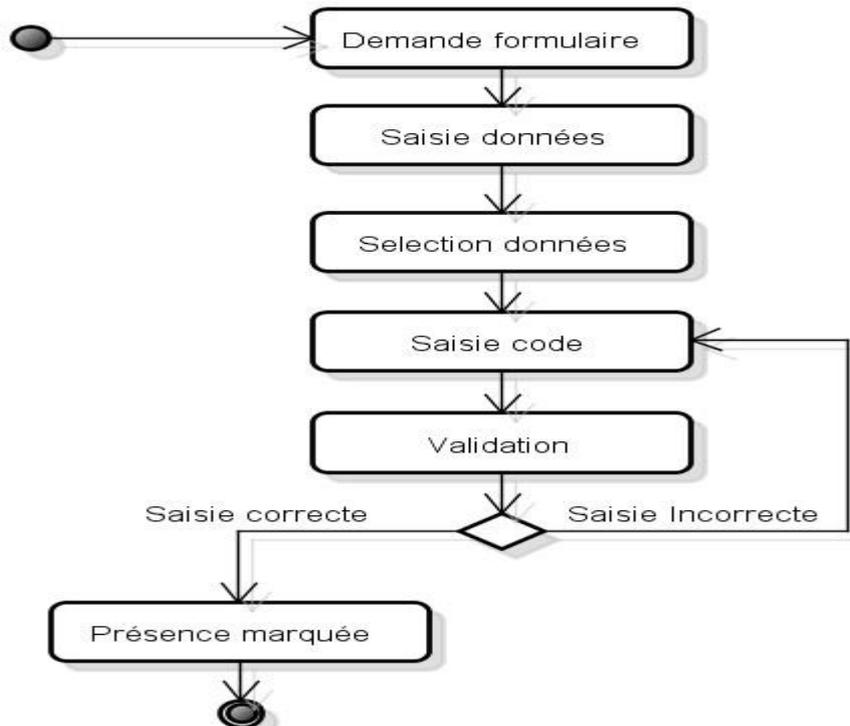
**Figure 3 : Diagramme de cas d'utilisation détaillée**

[6]

**I.2.3. Diagramme d'activités**

Le diagramme d'activité (Activity Diagram) fait partie des diagrammes comportementaux. Il est utilisé pour modéliser les aspects dynamiques d'un système. Il s'agit de représenter les opérations d'un processus et leurs conséquences sur les objets (logiciels ou matériels). La modélisation peut être utilisée pour décrire le déroulement d'un cas d'utilisation ou d'une méthode.

□ **Diagramme d'activité : Gestion présence**  
- Marquer présence

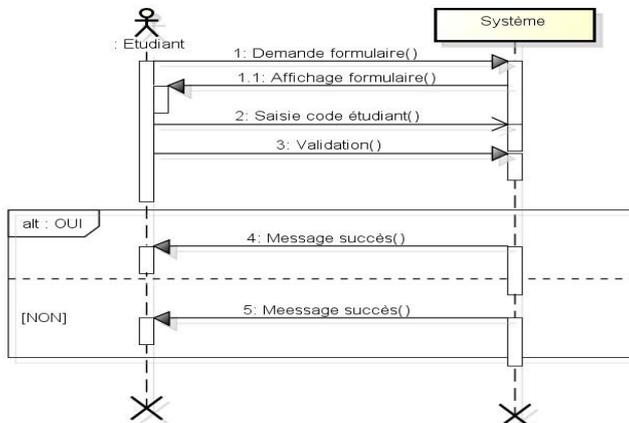


**Figure 4 : Diagramme d'activité marquer présence**

**I.2.3. Diagramme de séquences**

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but est de décrire comment se déroulent les interactions entre les acteurs ou objets.

□ **Diagramme de séquence : Gestion présence**  
- Marquer présence



**Figure 5 : Diagramme séquence Marquer présence**

**II. MODELISATION STATIQUE**

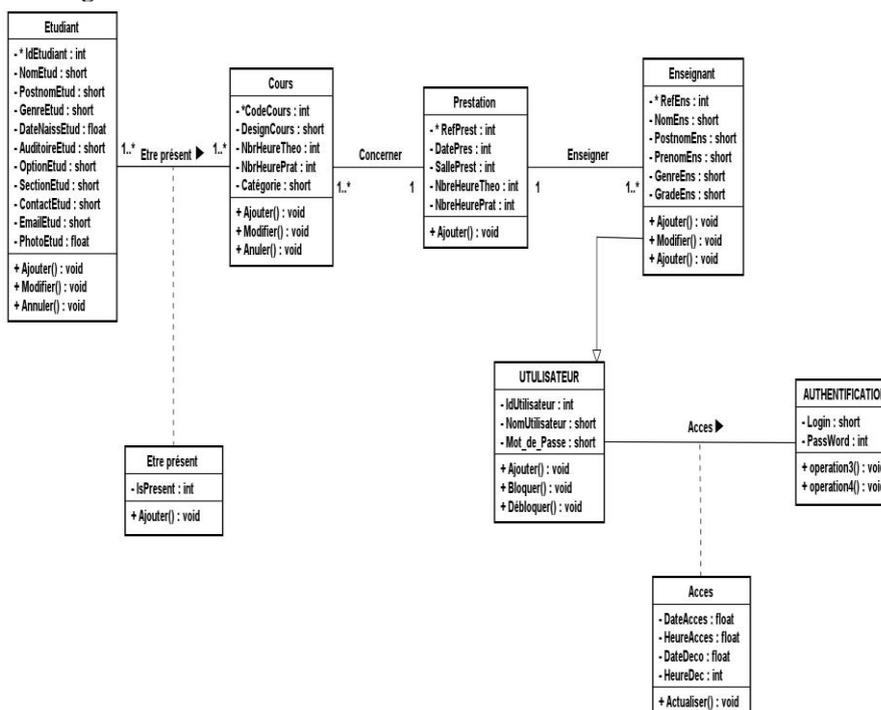
**II.1. Diagramme de classe**

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

Le diagramme de classes est généralement considéré comme le plus important dans le développement orienté objet. Il représente l'architecture conceptuelle du système interne : il décrit les classes que le système utilise ainsi que leurs liens, un héritage au niveau des classes afin de minimiser les interactions ou une agrégation entre deux classes. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Le diagramme de classe présenté ci-dessous est suivis d'un modèle relationnel pour esquisser ce qu'est implémenté logiquement dans la base des données utilisant un système de gestion de base des données relationnelles.

□ **Diagramme de classe global**



**Figure 6 : Diagramme de classe global**

### III. Test et présentation du projet

Cette partie est essentiellement consacrée à la présentation des principales interfaces du système sous forme de capture d'écrans. Chaque écran sera précédé ou suivi d'un commentaire pour décrire son fonctionnement.

**JeSuisPrésent** est un dispositif contrôleur de présence des étudiants, gère les feuilles d'émargement et les attestations pour les administrations. C'est un système automatisé qui au moyen de la programmation informatique des procédures de numérisation des feuilles de présence, génère la situation de l'étudiant durant l'enseignement.

#### 1. Formulaire d'authentification

Après avoir cliqué sur l'icône de l'application qui se trouve au bureau, le **formulaire d'authentification** va apparaître. Dans ce formulaire l'utilisateur sera obligé d'indiquer le couple login, c'est à dire le nom utilisateur et le mot de passe valide pour pouvoir se connecter dans le système. Cette interface se présente de la manière suivante.



Image 1 :Formulaire d'authentification

#### 2. Menu principale

A la fin de chargement des fichiers viendra l'apparition de **menu principal**. Dans cette fenêtre, l'utilisateur a l'obligation d'enregistrer soit un étudiant, un enseignant, un cours... et aussi il peut imprimer une liste de présences après prestation.

#### 3. Interface d'enregistrement étudiant

Comme nous l'avons dit dans le menu, surement dans cette interface l'utilisateur doit passer à l'enregistrement des étudiants dans la base de données. Après avoir cliqué sur le bouton « Etudiant » se trouvant au menu, l'image d'**interface étudiant** suivante est apparue.

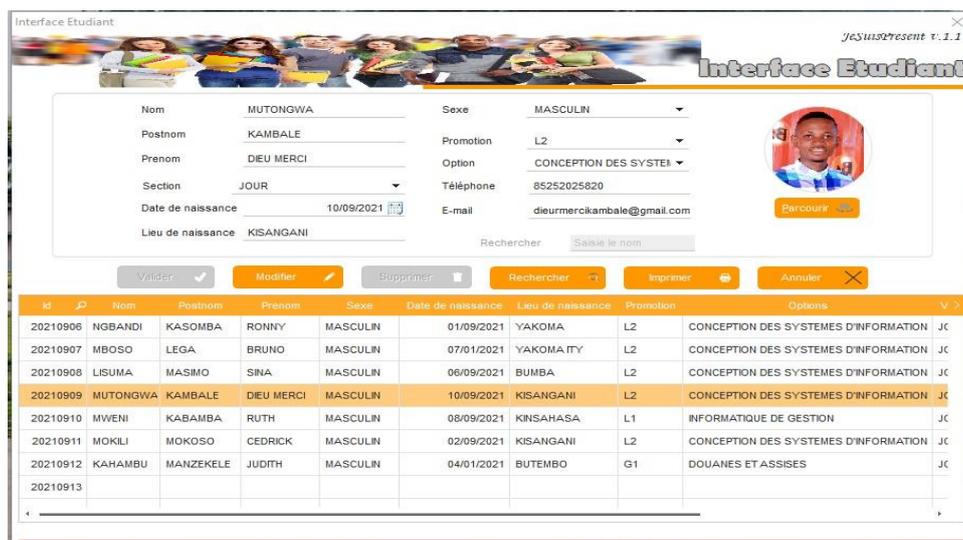


Image 2 : Interface étudiant 3. Interface d'enregistrement prestation A ce niveau, l'utilisateur devra effectuer l'enregistrement de la prestation journalière. Cette interface de prestation se présente comme suit.



**Image 3 : Interface prestation**

**4. Interface d'enregistrement présence**

A ce niveau, l'enseignant qui est considéré comme utilisateur devra indiquer la prestation du jour dans l'image 4, en suite les étudiants viendront les uns après les autres pour pouvoir signer leurs présences en renseignant leurs codes dans la zone de saisie « code étudiant » ici représenter dans l'image 5.



**Image 4 : Formulaire de présence**

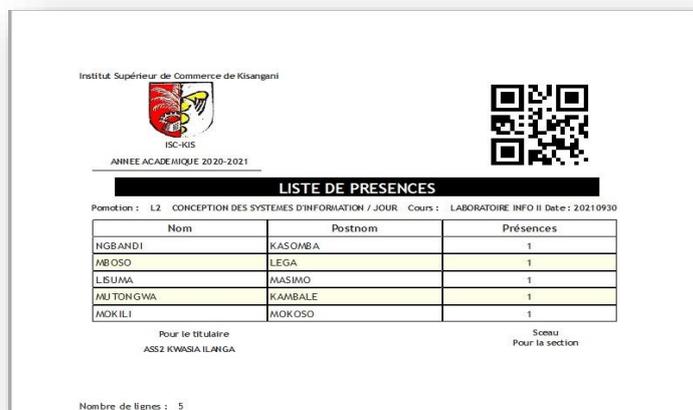


**Image 5 : Formulaire de présence**

**5. Présentation de quelques états de sortie**

Après avoir fourni plusieurs informations dans les différentes interfaces de l'architecture de notre application. Que ça soit au niveau de l'apparitorat par « l'appariteur » comme au niveau de l'auditoire par « l'enseignant » et « les étudiants » Voici donc quelques documents à imprimer.

- Liste des presences



**Image 6 : Etat de sortie liste de présences**

## Conclusion

Chaque début a une fin dit-on ! Nous voici donc au terme de la présente étude qui porte sur l'**implémentation d'un système client-serveur à deux niveaux de gestion intégrée des présences des étudiants au sein de l'institut supérieur de commerce de Kisangani.**

L'objectif de ce travail était d'examiner à partir du système de gestion existant dans

cette institution, voire la façon dont se passe le marquage de présences des étudiants au sein de l'institution.

Après l'analyse du système trouvé à l'Institut Supérieur de Commerce de Kisangani, nous avons remarqué quelques difficultés causées par le traitement manuel des informations. Nous avons en effet diagnostiqué le vrai problème du système trouvé avant de lui proposer une solution informatique pouvant apporter une certaine amélioration et d'éradiquer le problème trouvé. Nous avons dans un premier temps exposé l'organisme d'accueil, en faisant un aperçu sur la ville de Kisangani, ensuite nous avons entamé les différentes étapes du processus de développement UP, afin de mettre en œuvre notre solution à la problématique. Nous avons commencé par l'identification des besoins qu'on a ensuite modélisée à travers les différents diagrammes techniques et dynamiques.

Ce projet a fait l'objet d'une expérience intéressante, la réalisation de ce projet

nous a permis d'apprendre à nous organiser, d'améliorer nos connaissances et nos compétences dans le domaine de la programmation. Une tâche pas facile en regardant le nombre des modules inclus dans ce projet et le langage de programmation utilisé. Ainsi, nous avons utilisé : WLangage et son IDE : WinDev25 ; avec SGBD : HFSQL et le serveur HFSQL/Client afin de répondre aux besoins et aux attentes des utilisateurs et gestionnaires du système étudié.

## BIBLIOGRAPHIE

- [1]. Chamillard, G., Les cahiers du programmeur, éditions systèmes et informations/Editions Internet, Californie 2008.
- [2]. Christian SOUTOU, UML 2 pour les bases de données, Ed. EYROLLES, Paris, 2005.
- [3]. Roques, P., UML2 en action : de l'analyse des besoins à la conception, édition Euryolles, 4<sup>e</sup> édition, 2006.
- [4]. UML2 par la pratique, édition Euryolles, 5<sup>e</sup> édition, 2006.
- [5]. David Vandevelde, WinDev15, les fondamentaux du développement d'applications, Paris Edition 2010.
- [6]. □ Webographie
- [7]. <https://www.developpeurs.com>
- [8]. <https://www.openclassrooms.com>
- [9]. <https://www.Pcsoft.fr>