

EPIGRAPHE

« Chacun fait sa part Dieu pourvoit »

IN MEMORIUM

A mes regrettés frères et sœurs **Francine SIMBI, Xavier SIMBI, Gentil SIMBI** que la mort a retiré très tôt de notre affection et de ce monde, de là-haut, voici l'un de mes progrès ; les choses auraient été facile si vous étiez à mes côtés.

DEDICACE

Compte tenu des circonstances de peines que j'ai rencontrées tout au long de ma vie conjugale, sociale et estudiantine, j'ai toujours lutté pour arriver à décrocher un diplôme de graduat en informatique de gestion, chose accomplie par la grâce de **DIEU TOUT PUISSANT** et à travers notre **Seigneur JESUS CHRIST**.

Ainsi, avec joie, je dédie ce travail de fin de cycle de graduat à mon époux **RAOUL SHUNGU JEAN PIERE** pour son soutien financier, à mes parents **GAPARAY CLEOPHAS** et **FURAH BEATRICE**, à mes frères et sœurs **ERIC SIMBI**, **CYNTHIA SIMBI**, **CHARTE SIMBI** à mes enfants **GLORIA NIKITA SHUNGU** ET **ELVIS SHUNGU**, à **Mr MALE MAKANZU LEON** pour son soutien moral, à vous tous amis et collègues

REMERCIEMENTS

A terme de ce travail de fin d'étude du grade de graduat au sein de l'université libre de Kinshasa ULK en sigle, en s'orientant sur les conseils et directives du secrétariat académique ainsi que tout le corps administratif de l'ULK.

Nous sommes également redevables à travers la direction générale de l'ULK ainsi que son corps académique, qui nous ont facilité la tâche afin d'étudier au sein de cette université et d'arriver au bon port.

Le présent travail n'aurait pu être à son terme sans le soutien de nombreuses personnes à qui nous nous tenons à remercier nous nous excusons également auprès de celles qui ne retrouveront leurs noms sur cette page.

Nous tenons à remercier tout particulièrement le PROF () qui a bien voulu diriger ce travail.

Nous remercions notre co-directeur le **CT YVE NSILULU** pour l'encadrement et le soutien tout au long de ce travail ainsi que toutes les discussions techniques et enrichissantes.

Nos remerciements vont tout droit envers ma famille, celle de **DTN LUSADUSU MEDADARD**, la famille **MPUNGA ALEX**, la famille de **DR JULIUS ELUMBA** ainsi que à mes très chères amies **NANOUSHKA ELISABETHE LANGA** et **RUTH HORTENSE IYOLO LEGOMBASSIA**

SIMBI ALICE SHUNGU

INTRODUCTIN GENERALE

Depuis l'apparition de l'internet, le monde a connu beaucoup des changements dans l'évolution du travail et cela dans tous les domaines, la communication et le commerce en font partie ; le site web est l'un des moyens le plus utilisé pour faire connaître l'entreprise, la société, les marchandises, les célébrités, etc.

En ce qui concerne notre sujet : « **Conception et Réalisation d'un site web pour la gestion des transferts des joueurs de football, cas de la RDC** », les services du site nous seront très utiles dans le marketing et la visibilité des clubs ainsi que leurs joueurs, le site sera aussi une banque de toutes les informations concernant les joueurs et le Mercato, enfin le site web mettra fin à la tricherie qui se fait dans les transferts des joueurs.

0.1. PROBLEMATIQUE

Quand on parle de « problématique », on sous-entend l'ensemble des problèmes qui peuvent être révélés dans un cadre de gestion précis et auquel une ébauche des solutions vérifiables appelée « hypothèse » est proposée¹.

Selon le dictionnaire HACHETTE encyclopédie (2009), la problématique « est un ensemble des problèmes concernant un sujet » autrement dit ensemble des questions du départ en vue d'apporter des solutions éventuelles².

Parmi tous les problèmes qui entourent le marché des transferts des joueurs de football en RDC, nous avons remarqué que beaucoup des marchés ne sont pas fait d'une manière transparente (les informations ne sont pas sûres et fiables, les transferts frauduleux, etc.), et ceci a un impact très lourd sur un joueur en particulier et au club en général.

Ce qui nous a poussés à nous poser quelques questions suivantes en vue d'apporter une solution à l'égard de notre sujet :

- ❖ La gestion des joueurs en RDC est-elle efficace ?
- ❖ Les transferts des joueurs dans différentes situations ? : sous contrat, libre se font-ils sans problèmes ?
- ❖ Comment connaître la situation d'un joueur avant son transfert ?
- ❖ Ceux qui négocient les transferts des joueurs sont-ils habilités à le faire ?
- ❖ La conception et réalisation d'un site web pour la gestion des transferts des joueurs de football en RDC serait-elle la solution pour remédier à ces différents problèmes.

¹ NGAVO P., Cours inédit de Méthodes de recherche scientifique, G2 Info, ULK, Kinshasa, 2020-2021, p.11

² Dictionnaire HACHETTE encyclopédie, 2009.

0.2. HYPOTHESE

Comme nous l'avons étudié en deuxième année de graduat dans le cours des Méthodes de recherche scientifique, l'hypothèse est une série des réponses provisoires ou supposées mais vraisemblables au regard des questions soulevées au cours de notre problématique³.

En ce qui nous concerne, après analyse nous avons remarqué que la gestion des transferts des joueurs n'est pas efficace :

- Il y a une forte tricherie dans les différents transferts surtout en matière des documents, il y a des agents qui les falsifient sans tenir compte de la carrière du joueur, voire même la corruption ;
- Il y a un manque terrible d'informations crédibles concernant un joueur, les quelques informations que l'on puisse avoir proviennent des secrétaires des clubs qui peuvent vous livrer des informations erronées ;
- Beaucoup de secrétaires des clubs, managers, etc. ne maîtrisent pas des lois sportives régissant les transferts des joueurs.

En se référant aux différents éléments cités ci-haut, nous croyons qu'un site web sera d'une importance capitale pour la gestion des transferts des joueurs, il nous permettra d'avoir les informations concernant les transferts des joueurs à temps réel et de connaître la situation de chaque joueur.

0.3. CHOIX ET INTERET DU SUJET

Nous avons choisi ce sujet pour apporter notre pierre dans la construction et la revalorisation de ce métier de manager ou agent de joueur que nous aimons tant et aussi parce qu'il y a peu de site web qui fournissent des informations sur le management des footballeurs au niveau national et international.

Le présent travail nous permettra de répondre aux exigences académiques pour sanctionner la fin de notre premier cycle d'études universitaires et servira d'un outil didacticiel pour les générations futures.

³ NGAVO P., Op.cit., p.12

0.4. DELIMITATION DU TRAVAIL

Tout travail scientifique doit avoir une délimitation spatio-temporelle :

4.1. DANS LE TEMPS

Nous signalons que les recherches menées pour ce travail ont couvert la période allant de

Décembre 2020 à Janvier 2022.

4.2. DANS L'ESPACE

Ce travail se limitera à la réalisation d'un site web pour les transferts des joueurs évoluant en RDC et ce site est aussi conçu pour la promotion des joueurs performants au niveau local (RDC).

0.5. METHODES ET TECHNIQUES UTILISEES

Toute œuvre scientifique est utile d'être bien faite, résolument avec un certain ordre. Pour la réaliser, c'est toujours important de faire le choix des méthodes et techniques à utiliser afin de produire un bon résultat.

5.1. METHODES UTILISEES

Pour réaliser ce travail, nous avons fait appel aux méthodes suivantes :

A. Méthode analytique:

Elle nous a permis d'analyser les informations récoltées concernant les transferts des joueurs.

B. Méthode descriptive:

Elle nous a permis de décrire les informations récoltées concernant les transferts des joueurs.

C. Méthode de MERISE :

C'est une méthode de conception, de développement et de réalisation des projets informatiques, dont le but est d'arriver à concevoir un système d'information informatisé.

Elle est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques⁴.

⁴ DIGALLO Frédéric, Méthodologies des systèmes d'information : la méthode MERISE, CNAM, Paris, Juillet, 2001

D. Le langage UML

UML (*Unified Modeling Language*, que l'on peut traduire par "*langage de modélisation unifié*") est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en termes de modélisation objet.



Figure 1 : UML.

5.2. TECHNIQUES

La technique est l'ensemble de procédés employés pour produire une œuvre en obtenant le résultat escompté. Autrement dit, c'est comme « un moyen d'atteindre un but, un ensemble d'outils mis à la disposition de la recherche et organisés par la méthode pour atteindre un objectif précis »⁵.

Pour appuyer notre méthode, les techniques ci-après nous ont été utiles :

A. Technique d'interview : elle constitue l'ensemble d'investigations menées auprès des différents postes concernés par l'étude. Ces investigations sont formulées par des questions dont les réponses constituent les données.

B. Technique d'analyse documentaire : elle nous a permis de recueillir les données sur base des documents mis à notre disposition et aux différentes consultations (Textes légaux, contrats et articles relatifs à notre sujet).

0.6. SUBDIVISION DU TRAVAIL

Hormis l'introduction et la conclusion générale, ce travail s'organise en 3 chapitres :

- ❖ Chapitre I. Concepts théoriques de base
- ❖ Chapitre II. Conception du site web
- ❖ Chapitre III. Réalisation du site web

⁵ NGAVO P., op.cit., p. 16

CHAPITRE I. CONCEPTS THEORIQUES DE BASE

I.1. CONCEPTS INFORMATIQUES

1.0. Site web : selon le dictionnaire français LAROUSSE, est un ensemble de web accessibles via internet sur un serveur identifié par une adresse.

1.0.1. SITE : selon le dictionnaire LE ROBERT est un paysage du point de vue de l'esthétique, du pittoresque.

1.0.2. SITE WEB : Techniquement un site web est un fichier ou ensemble des fichiers hébergés sur un serveur et accessible via internet. Un site web est avant tout un ensemble des pages web reliées entre elles qui partager un nom de domaine. Chaque page d'un site fournit des liens explicites qui permettent à l'utilisateur entre les pages d'un site.

1.0.3. SERVEUR : Un serveur informatique bien évidemment le mot serveur a d'autres signification, est un dispositif informatique qui offre des services à un ou plusieurs clients notamment l'accès aux informations de la world wide web, le courrier électronique ; le commerce électronique, le stockage en base des données. Nous pouvons citer quelques différents types de serveurs :

Serveur de fichiers : qui hébergent et diffusent des fichiers que peuvent une multitude d'utilisateurs :

- SERVEUR D'IMPRESSION
- SERVEUR D'APPLICATION
- SERVEUR DE MESSAGERIE
- SERVEUR DE WEB
- SERVEUR DE BASE DES DONNEES.

1.0.4. CLIENT : Un processus demandant l'exécution d'un message contenant la description d'opération par message retour.

1.0.5. BDD (Base des données) Une base des données est une collection d'informations structurées, stockées sur un support informatique accessible par l'ordinateur en vue d'une utilisation simultanée par plusieurs utilisateurs. Une base des données doit respecter ces trois principales ci-après :

Une base des données doit être **Exhaustive** c'est-à-dire qu'il doit avoir toutes les informations possibles en rapport avec un domaine de gestion bien précis.

Une base des données doit être **Structurée** enfin de permettre une bonne manipulation.⁶

La non redondance : une base des données bien conçue ne doit pas permettre la répétition des informations en termes d'enregistrement ou d'occurrence.

1.0.5 : SGBD : Système de gestion des bases des données est un ensemble des fichiers qui permet de manipuler le contenu des bases des données. Il sert à effectuer les opérations telles que Rechercher, Ajouter, Modifier ou Supprimer des enregistrements, manipuler les index, créer des bases des données ... Les principaux SGBD connus sur le marché sont : IMB, OB2, ORACLE DATA BASE, MYSQL, SQL serveur, ACCESS pour ne citer que cela.

I.1.1. Généralités sur le web

Le web fut inventé en 1989 par Tim Berners-Lee et une équipe de recherche au **CERN** (Centre Européen pour la recherche Nucléaire). C'est un système hypertexte qui permet de lier les documents par des hyperliens. Le web est possible grâce aux réseaux internet qui relient entre eux les ordinateurs. Certains ordinateurs hébergent des pages web : les serveurs et chaque page web a une adresse URL (Uniform resource locator). Le client qui est mon ordinateur peut demander qu'on lui transmette des pages web à partir de leur URL Grâce au protocole http (hypertexte Transfer Protocol).

⁶ Note de cours de TBDD par le professeur ISAKATONGA Justin

I.1.2. LE ROLE DU NAVIGATEUR

Le navigateur permet de traduire toutes les informations contenues dans les fichiers HTML, CSS et JS et le résultat s'affiche à l'écran avec lequel l'utilisateur peut interagir et cela Grâce à la combinaison de CTRL+U.

I.1.3. Notions sur les sites web

Les sites web sont ensemble des pages web et des ressources inter reliées entre elles par des liens hypertextes, auxquelles l'internaute peut accéder par une adresse web appelée URL. Un site web ou site internet englobe des textes et multimédia, il est hébergé sur un serveur web auquel on peut accéder à travers le réseau internet ou intranet. Comme je l'ai expliqué dans l'introduction, l'internet est devenu l'outil le plus utilisé au monde qui relie plusieurs internautes à travers le monde dépassant de loin les medias classiques tels que la télévision. Une page web contient des informations qui permettent de communiquer et de partager des ressources avec les gens qui sont connectes sur l'internet et ce partout dans tous les domaines (le commerce, la communication, le tourisme, l'enseignement...) pour faire connaitre leurs services, adresses, les prix des marchandises, la promotion de marque pour affirmer sa présence sur le réseau internet. Il faut noter qu'il y a une diversité des sites web :

- ❖ Les sites e-commerce qui pour principal objectif la vente en ligne.
- ❖ Les sites vitrines qui sont majoritairement utilisés pour présenter et exposer les services d'une organisation ou une marque.
- ❖ Les sites Institutionnels qui présentent une organisation et ses valeurs à travers la description de son activité.
- ❖ Les sites de portails qui proposent des services de messageries d'actualités....
- ❖ Les sites personnels quant eux réalisent au profit des particuliers qui désirent partager en ligne leur passion pour un sujet précis.

I.1.3. Notions sur la programmation web

La programmation web c'est l'écriture du web au balisage et au codage impliqués dans le développement web. Les langages les plus utilisés pour la programmation web sont XML, HTML, JAVASCRIPT, PERL5 et PHP. Il faut signaler qu'il y a une différence entre la programmation web et une simple programmation qui nécessite des connaissances interdisciplinaires sur le domaine⁷ d'application, les scripts clients et serveur et la technologie de base des données. La programmation web peut être brièvement classée en codage client et serveur⁸.

a) Le côté client : le côté client a besoin d'une programmation liée à l'accès aux données des utilisateurs et à la fourniture d'informations. Il doit également s'assurer qu'il y a suffisamment de plug-ins pour enrichir l'expérience utilisateurs graphique y compris de mesures de sécurité.

b) Le côté serveur : le côté serveur a besoin d'une programmation principalement liée à la récupération des données, à la sécurité et aux performances. Certains outils comme ASP, LOTUS NOTES ; PHP, JAVA et MYSQL sont utilisés.

- **LE CLIENT / SERVEUR CLASSIQUE :** ça se passe entre le client et serveur. Le client envoie une requête au serveur qui à son tour analyse et envoie la réponse au client.
- **LE CLIENT / SERVEUR RICHE :** Le client envoie une requête à *API (Application Programming Interface)* qui envoie à son tour au serveur pour l'analyse et envoie la réponse au composant instance (composant) qui à son tour renvoie à API pour envoyer la réponse au client. Avec le client/serveur riche il y a les intermédiaires.

I.1.4. Notions sur la base de données et le S.G.B.D.

BDD (Base des données) Une base des données est une collection d'informations structurées, stockées sur un support informatique accessible par l'ordinateur en vue d'une utilisation simultanée par plusieurs utilisateurs.

⁸ Note de cours M.A.I par CT Yves Nsilulu

Une base des données doit respecter plusieurs principes nous pouvons citer entre autre :

I.1.4.1. LES CARACTERISTIQUES D'UNE BASE DE DONNEES

Une base des données doit être **Exhaustive** c'est-à-dire qu'il doit avoir toutes les informations possibles en rapport avec un domaine de gestion bien précis.

Une base des données doit être **Structurée** enfin de permettre une bonne manipulation.

- **La non redondance** : une base des données bien conçue ne doit pas permettre la répétition des informations en termes d'enregistrement ou d'occurrence.
- **Utilité d'une base de données** : Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses. Une base de données peut être **locale**, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien **répartie**, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau.

L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.

- **La gestion d'une base de données** : Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion de la base de données se fait grâce à un système appelé **SGBD** (Système de Gestion de Bases de Données) ou en anglais **DBMS** (Data base Management System). Le SGBD est un ensemble de services (applications logicielles) permettant de gérer les bases de données, c'est-à-dire:
 - Permettre l'accès aux données de façon simple
 - Autoriser un accès aux informations à de multiples utilisateurs
 - Manipuler les données présentes dans la base de données (insertion, suppression, modification).

SGBD : Système de gestion des bases des données est un ensemble des fichiers qui permet de manipuler le contenu des bases des données. Il sert à effectuer les opérations telles que Rechercher, Ajouter, Modifier ou Supprimer des enregistrements, manipuler les index, créer des bases des données ... Les principaux SGBD connus sur le marché sont : IMB, OB2, ORACLE DATA BASE, MySQL, SQL serveur, ACCESS pour ne citer que cela.

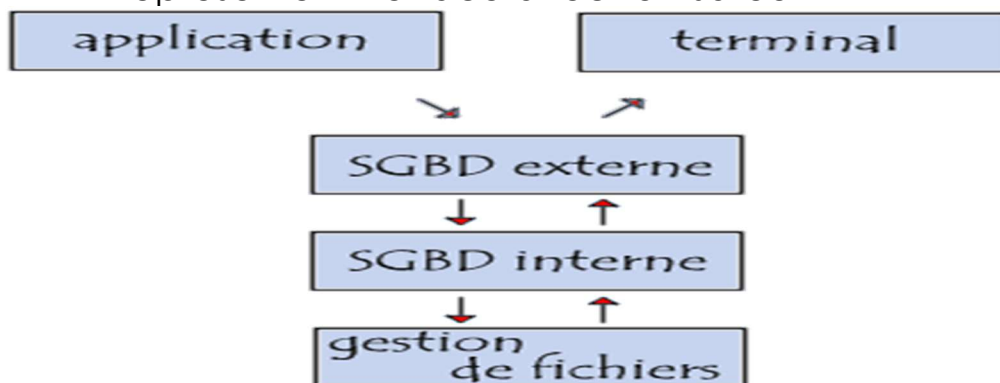
- **Caractéristiques d'une base de données** : L'architecture à trois niveaux définie par le standard ANSI/SPARC permet d'avoir une indépendance entre les données et les traitements. D'une manière générale un SGBD doit avoir les caractéristiques suivantes:
 - o **Indépendance physique**: le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données n'apparaissent pas pour l'utilisateur, il s'agit simplement d'une structure transparente de représentation des informations ;
 - o **Indépendance logique**: le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c'est-à-dire que l'administrateur de la base doit pouvoir la faire évoluer sans que cela gêne les utilisateurs ;
 - o **Manipulabilité**: des personnes ne connaissant pas la base de données doivent être capables de décrire leurs requêtes sans faire référence à des éléments techniques de la base de données ;
 - o **Rapidité des accès**: le système doit pouvoir fournir les réponses aux requêtes le plus rapidement possibles, cela implique des algorithmes de recherche rapides ;
 - o **Administration centralisée**: le SGBD doit permettre à l'administrateur de pouvoir manipuler les données, insérer des éléments, vérifier son intégrité de façon centralisée.
- **Limitation de la redondance**: le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d'éviter d'une part un gaspillage d'espace mémoire mais aussi des erreurs
- **Vérification de l'intégrité**: les données doivent être cohérentes entre elles, de plus lorsque des éléments font références à d'autres, ces derniers doivent être présents.

- **Partageabilité des données:** le SGBD doit permettre l'accès simultané à la base de données par plusieurs utilisateurs

Sécurité des données: le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs

Le SGBD peut se décomposer en trois sous-systèmes:

- Le système de gestion de fichiers (SGF):
Il permet le stockage des informations sur un support physique
- Le SGBD interne:
Il gère l'ordonnancement des informations
- Le SGBD externe:
Il représente l'interface avec l'utilisateur



I.5. NOTION SUR UML.

5.1.1. Introduction

Le Langage de Modélisation Unifié est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en termes de modélisation objet. UML (*Unified Modeling Language*, que l'on peut traduire par

5.1.1.2. Notion d'objet

Elle consiste à modéliser informatiquement un ensemble d'éléments d'une partie du monde réel (que l'on appelle *domaine*) en un ensemble d'entités informatiques. Ces entités informatiques sont appelées *objets*. Il s'agit de données informatiques regroupant les principales caractéristiques des éléments du monde réel (taille, la couleur...). La difficulté de cette modélisation consiste à créer une représentation abstraite, sous forme d'objets, d'entités ayant une existence matérielle (chien, voiture, ampoule, ...) ou bien virtuelle (sécurité sociale, temps, ...).

5.1.1.3. La Méthode Objet

Elle consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse, sans se préoccuper de l'implémentation, ce qui signifie *indépendamment d'un langage de programmation*. Il s'agit donc de déterminer les objets présents et d'isoler leurs données et les fonctions qui les utilisent. Pour cela des méthodes ont été mises au point. Entre 1970 et 1990, de nombreux analystes ont mis au point des approches orientées objets, si bien qu'en 1994 il existait plus de 50 méthodes objet. Toutefois seules 3 méthodes ont véritablement émergé :

- La méthode **OMT** de *RUMBAUGH*
- La méthode **BOOCH'93** de *BOOCH*
- La méthode **OOSE** de *JACOBSON* (*Object Oriented software engineering*)

À partir de 1994, rumbaugh et booch (rejoints en 1995 par jacobson) ont unis leurs efforts pour mettre au point la méthode unifiée (*unified method 0.8*), incorporant les avantages de chacune des méthodes précédentes.

La méthode unifiée à partir de la version 1.0 devient **UML** (*unified modeling language*), une notation universelle pour la modélisation objet.

UML 1.0 est soumise à l'OM. (*object management group*) en janvier 1997, mais elle ne sera acceptée qu'en novembre 1997 dans sa version 1.1, date à partir de laquelle UML devient un standard international.

Voici le récapitulatif des évolutions de ce langage de modélisation :

- En 1995: méthode unifiée 0.8 (intégrant les méthodes booch'93 et omt)
- En 1995: UML 0.9 (intégrant la méthode oose)
- En 1996: UML 1.0 (proposée à L'OMG)
- En 1997: UML 1.1 (standardisée par l'OMG)
- En 1998: UML 1.2
- En 1999: UML 1.3
- En 2000: UML 1.4
- En 2003: UML 1.5

Cette méthode représente un moyen de spécifier, représenter et construire les composantes d'un système informatique.

5.1.1.6. Caractéristiques d'UML

a) Le Meta Model

UML est un moyen d'exprimer des modèles objet en faisant abstraction de leur implémentation, c'est-à-dire que le modèle fourni par UML est valable pour n'importe quel langage de programmation. UML est un langage qui s'appuie sur un méta modèle, un modèle de plus haut niveau qui définit les éléments d'UML (les concepts utilisables) et leur sémantique (leur signification et leur mode d'utilisation).

Le méta modèle permet de se placer à un niveau d'abstraction supérieur car il est étudié pour être plus générique que le modèle qu'il permet de construire.

Le méta-modèle d'UML en fait un langage formel possédant les caractéristiques suivantes:

- un langage sans ambiguïtés
- un langage universel pouvant servir de support pour tout langage orienté objet
- un moyen de définir la structure d'un programme
- une représentation visuelle permettant la communication entre les acteurs d'un même projet
- une notation graphique simple, compréhensible même par des non informaticiens

Le méta model permet de donner des bases solides et rigoureuses à ce langage graphique, dont les représentations graphiques ne sont là que pour véhiculer des concepts de réalisation.

b) UML comme model de visualisation d'un système

UML offre une manière élégante de représenter le système selon différentes vues complémentaires grâce aux diagrammes.

5.1.2. Modélisation avec UML

a) Qu'est-ce qu'un model

Le modèle est une représentation simplifiée d'un problème. Grâce au modèle il est possible de représenter simplement un problème, un concept et le simuler. La modélisation comporte deux composantes:

- L'analyse, c'est-à-dire l'étude du problème

La conception, soit la mise au point d'une solution au problème.

5.1.3. Modélisation des classes et objets

a) Modélisation d'un objet

La modélisation objet consiste à créer une représentation abstraite, sous forme d'objets, d'entités ayant une existence matérielle (arbre, personne, téléphone, ...) ou bien virtuelle (sécurité sociale, compte bancaire, ...). Un objet est caractérisé par plusieurs notions:

- **les attributs** (on parle parfois de propriétés): il s'agit des données caractérisant l'objet. ce sont des variables stockant des informations d'état de l'objet
- **les méthodes** (appelées parfois fonctions membres): les méthodes d'un objet caractérisent son comportement, c'est-à-dire l'ensemble des actions (appelées opérations) que l'objet est à même de réaliser. Ces opérations permettent de faire réagir l'objet aux sollicitations extérieures (ou d'agir sur les autres objets). De plus, les opérations sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs, ou bien les modifier
- **l'identité**: l'objet possède une identité, qui permet de le distinguer des autres objets, indépendamment de son état. On construit généralement cette identité grâce à un identifiant découlant naturellement du problème (par exemple un produit pourra être repéré par un code, une voiture par un numéro de série, ...)

UML propose une manière de représenter les objets de façon graphique, sous forme de **rectangle, dans lequel le nom de l'objet est souligné**.

A rectangular box with a light blue background and a black border. Inside the box, the text 'objet 1' is written in a black, handwritten-style font. The text is underlined with a thin black line.

A rectangular box with a light blue background and a black border. Inside the box, the text 'objet 2' is written in a black, handwritten-style font. The text is underlined with a thin black line.

A rectangular box with a light blue background and a black border. Inside the box, the text 'objet 3' is written in a black, handwritten-style font. The text is underlined with a thin black line.

b) Les attributs d'un objet

Les attributs (propriétés) d'un objet représentent ses caractéristiques. L'ensemble des valeurs des attributs d'un objet constituent son état.

c) Modélisation d'une classe

On appelle classe la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui composeront un objet. Un objet est donc "issu" d'une classe, c'est le produit qui sort d'un moule. En réalité on dit qu'un objet est une **instanciation** d'une classe, c'est la raison pour laquelle on pourra parler indifféremment d'objet ou d'instance (éventuellement d'occurrence).

* Une classe est composée:

- D'attributs: il s'agit des données, dont les valeurs représentent l'état de l'objet
- Les méthodes : il s'agit des opérations applicables aux objets

Si on définit la classe *voiture*, les objets *peugeot 406*, *volkswagen golf* seront des instanciations de cette classe. Il pourra éventuellement exister plusieurs objets *peugeot 406*, différenciés par leur numéro de série.

Mieux: deux instanciations de classes pourront avoir tous leurs attributs égaux sans pour autant être un seul et même objet (c'est la différence entre *état* et *identité*). C'est le cas dans le monde réel, deux t-shirts peuvent être strictement identique (avoir le même état) et pourtant ils sont distincts (ils ont chacun leur identité propre). D'ailleurs en les mélangeant il serait impossible de les distinguer...

Une classe se représente avec UML sous forme d'un rectangle divisé en trois sections. Le premier contient le nom donné à la classe (non souligné). Les attributs d'une classe sont définis par un nom, un type (éventuellement une valeur par défaut, c'est-à-dire une valeur affectée à la propriété lors de l'instanciation) dans le second compartiment. Les opérations sont répertoriées dans le troisième volet du rectangle.

d) Visibilité des attributs

L'un des principaux concepts du paradigme objet est. L'encapsulation est un mécanisme consistant à rassembler les données et les méthodes au sein d'une structure en cachant l'implémentation de l'objet, c'est-à-dire en empêchant l'accès aux données par un autre moyen que les services proposés. L'encapsulation permet donc de garantir l'intégrité des données contenues dans l'objet.

En effet, la programmation orientée objet permet de cacher l'implémentation d'un objet en ne lui permettant d'accéder aux attributs uniquement par l'intermédiaire de méthodes créées à cet effet (on parle d'interface).

Il est ainsi possible de définir des niveaux d'encapsulation, afin de définir le type de classe ayant accès aux attributs.

On parle de niveau de visibilité des éléments de la classe. Ces niveaux de visibilité définissent les droits d'accès aux données selon que l'on y accède par une méthode de la classe elle-même, d'une classe, ou bien d'une classe quelconque. il existe trois niveaux de visibilité

- **public**: les fonctions de toutes les classes peuvent accéder aux données ou aux méthodes d'une classe définie avec le niveau de visibilité *public*. il s'agit du plus bas niveau de protection des données
- **protégée**: l'accès aux données est réservé aux fonctions des classes ; c'est-à-dire par les fonctions membres de la classe ainsi que des classes
- **privée**: l'accès aux données est limité aux méthodes de la classe elle-même. il s'agit du niveau de protection des données le plus élevé

La notation UML permet de représenter le niveau de visibilité des attributs de façon graphique en faisant précéder le nom de chaque attribut par un caractère représentant la visibilité:

- + défini un attribut public
- # défini un attribut protégé
- - défini un attribut privé

* LES DIAGRAMMES

- **Le diagramme de classe** : structure statique (classe et association).Il modélise le comportement et permet de capturer les exigences du système ; il définit également les interactions entre les acteurs et le système.
- **Le diagramme d'objet** : instance du diagramme de classe (objet et lien)
- **Le diagramme de cas d'utilisation** : fonction du point de vue des utilisateurs
- **Le diagramme de comportement**
- **Le diagramme de séquence** : aspect temporelle des interactions entre des objets (séquence d'évènement). C'est une représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Il décrit comment et dans quel ordre les objets fonctionnent ensemble.
- **Le diagramme de collaboration** : aspect spécial des interactions entre les objets (relativement à leurs rôles et aux liens qu'ils ont entre eux)

- **Le diagramme d'activité** : comportement d'une opération en termes d'action et activité
- **Le diagramme d'état-transition** : comportement dynamique des objets (changement d'état en réaction à des évènements)
- **Le diagramme de réalisation** : unité de travail
- **Le diagramme de composant** : composants logiciels réalisant l'application (code sources, bibliothèque, dépendance, etc.)
- **Le diagramme de déploiement** : repartitions des composants logiciels sur des matériels. Il modélise l'architecture physique d'un système, il affiche les relations entre les composants logiciels et matériels du système.
- **Le diagramme de paquetage** : permet de structurer le modèle, la réutilisation, d'organiser l'équipe de développement de jeter des bases de la gestion de la configuration.

QUELLQUES NOTIONS SUR :

- **Entité** : est un objet, sujet, une notion en rapport avec le domaine d'activité pour laquelle la base de données est utilisée
- **Attribut** : est une caractéristique d'une entité susceptible d'être enregistré dans une base de données.
- **Enregistrement** : est une données composite qui comporte plusieurs champs dans chacun des quels est enregistrée une donnée.
- **Association** : désigne les liens qui existent entre différentes entités.
- **Cardinalité** : est le nombre des fois minimum et maximum qu'une entité participe dans une association :⁹

I.2. CONCEPTS RELATIFS AU SUJET

Nous faisons une étude sur la gestion des transferts des joueurs de football en RDC et dans une limitation de temps d'une période de deux ans entre 2020 et fin janvier 2022. Ici comme ailleurs le mercato se fait 3 fois par an (année sportive) ce qui signifie que dans une fédération de football bien organisée il est facile de connaître le nombre des joueurs qui ont été vendus en une saison sportive. Bien que le football soit une discipline qui draine un nombre impressionnant de fanatique c'est pour cette raison il est classée mondialement numéro 1, le football a ses termes appropriés qui ne sont pas compréhensibles à tout le monde qui s'intéresse à ladite discipline que nous avons jugé bon d'expliquer pour la bonne compréhension à celui qui prendra son temps pour lire notre ouvrage.

I.2.1. Théories sur les transferts des joueurs

Contrairement à ce que l'on pourrait penser, la plupart des joueurs transférés ne sont pas vendus et n'engendrent pas des frais d'acquisition.

Qui sont vendus ? Sont vendus les joueurs transférés depuis un autre club vers un autre alors qu'ils étaient liés à leurs club d'origines par un contrat toujours en vigueur. Dans les autres cas ils étaient en fin de contrat ou en prêt.

NB : Un transfert n'est pas forcément une vente.¹⁰

Un joueur en fin de contrat n'est pas un joueur gratuit.¹¹

Un joueur en fin de contrat avec un club A, qui signe un nouveau contrat avec un club B entraîne quand même des dépenses :¹²

Prime à¹³ la signature pour le joueur ;

Prime à la signature pour l'agent du joueur.

*** Le rôle des agents**

En tant que gestionnaire des intérêts financiers d'un ou plusieurs joueurs, un agent a un rôle particulièrement important dans les transferts. Si une vente est effectuée 10% du montant du transfert reviendra à l'agent. Il s'agit donc d'une activité hautement rémunératrice au très haut niveau.

¹⁰ Articles des journaux, Internet

¹¹ Article des journaux, télévision et internet.

- **COMMENT SE PASSE UN TRANSFERT ?**

Le marché des transferts que l'on appelle aussi mercato se déroule en deux phases. La première pendant l'été, entre deux saisons et la deuxième à la moitié du championnat après les fêtes de fin d'année. En dehors de ces deux périodes un joueur ne peut être transféré à un autre club.

QUI TOUCHE L'ARGENT D'UN TRANSFERT ?

Si le joueur touche parfois aussi de l'argent lors de son transfert sous le couvert d'une « prime à la signature de contrat », les principaux bénéficiaires sont les clubs vendeurs et, de plus en plus les agents des joueurs appelés Managers.

I.2.2. Définitions de quelques concepts liés au sujet

- A. Conception :** La conception d'un projet informatique est un ensemble d'activités pour résoudre un problème précis tout en respectant les étapes en vue d'obtenir un résultat souhaité.
- B. Réalisation :** La réalisation est un accomplissement d'un projet.
- C. Gestion :** La gestion consiste à gérer ou prévoir (les buts et les moyens), organiser (les moyens et les méthodes permettant d'atteindre les objectifs).
- D. Joueurs :** Le joueur est une personne qui a la passion du jeu qui joue actuellement ou habituellement à un jeu.
- E. Transfert :** en football un transfert correspond au changement de club d'un footballeur professionnel. Un transfert est généralement le rachat du contrat d'un joueur, mais peut prendre d'autres formes comme un prêt avec obligation d'achat, un prêt avec option d'achat ou un prêt sans option d'achat.
- F. AGENT DE FOOTBALL :** Dans le football un agent appelé aussi le « Manager » est une personne chargée des affaires des joueurs dans un club c'est à dire l'achat et la vente des joueurs
- G. Mercato :** Le Mercato est simplement un marché des transferts des joueurs.

Le marché des transferts aussi appelé MERCATO (Italianisme signifiant « marché ») est la période pendant laquelle les clubs professionnels de football sont autorisés à transférer ou à prêter leurs joueurs à d'autres clubs.

CHAPITRE II. CONCEPTION DU SITE WEB

Section 1. Modélisation des données.

INTRODUCTION

La modélisation Est la conception d'un model.la modélisation de données est l'analyse et la conception de l'information contenue dans le système. Il s'agit essentiellement d'identifier les entités logiques et les dépendances logiques entre ces entités.

La modélisation des données en informatique est en réalité un processus de description de la structure, des associations, des relations et des impératifs liés à des données disponibles.

Elle permet de fixer des normes tout en codant des modèles de gestion des données dans une organisation.

1.1. ETAPES DE LA MODELISATION.

Dans le cadre de notre projet, nous utiliserons le modèle en cascade qui est un standard Américain DoD-STD-2167. Ce dernier repose sur une forme extrêmement simplifiée du modèle de gestion développé par **Royce**.

En pratique, **plusieurs versions du modèle en cascade** sont utilisées. Les modèles les plus courants divisent les processus de développement en cinq phases. Les phases 1, 2 et 3 définies par Royce sont parfois regroupées en une seule et même phase, qualifiée d'analyse des besoins.

1. Analyse : Planification, analyse et spécification des besoins;
2. Conception : conception et spécification du système ;
3. Implémentation : programmation et tests des modules ;
4. Test : intégration du système, tests du système et l'intégration ;
5. Exploitation : livraison, maintenance, amélioration.

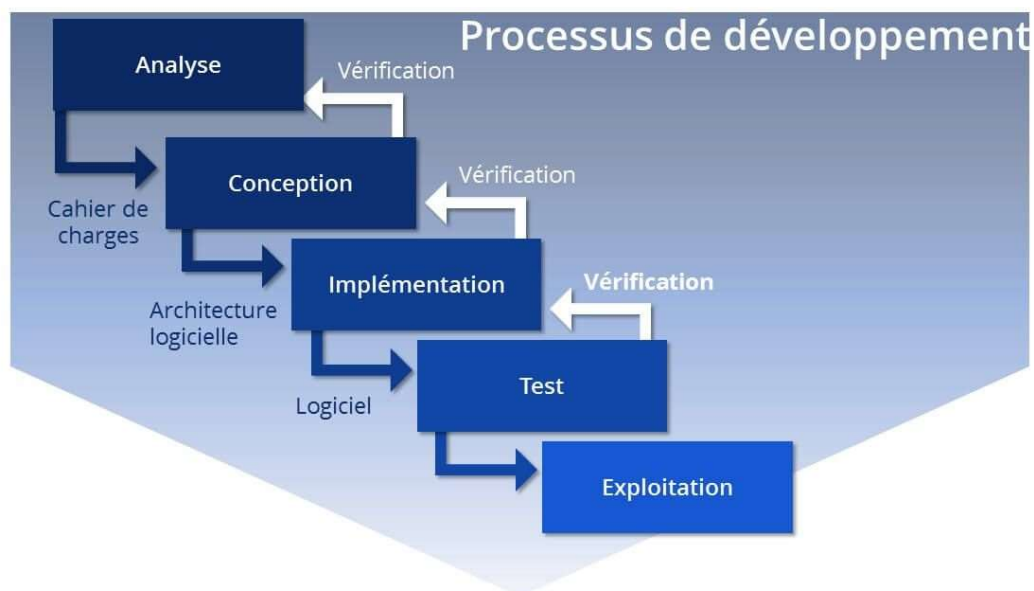


Figure 2 : Processus de développement.

- **Type de modélisation :**

Il y a plusieurs méthodes de modélisation des données mais les plus utilisées sont :

RM/t, Merise, les diagrammes de BACHMAN, L'approche règles de gestion, la modélisation Rôle d'objet (Object role modeling), la modélisation objet-relation et UML ; pour notre travail nous avons choisi d'utiliser la méthode UML.

1.2. LE BUT DE LA MODELISATION DE DONNEES

Modéliser des données consiste à mettre au point un ensemble d'équations ou des règles pour décrire un phénomène de façon reproductible et simulable. Le modèle issu de la modélisation sert à prédire le comportement d'un système en fonction de sollicitations connues.

1.2.1. LA MODELISATION

1.2.1.1. PRESENTATION DES DIAGRAMMES.

a) La narration

Après avoir apprécié les prestations sur un joueur, le club intéressé contacte le club cédant via leurs managers respectifs, le joueur passe une série de tests (physique et médicaux). Si ce dernier réussit les tests les clubs entament les négociations sur le prix, une fois trouve l'accord, le joueur signe le contrat moyennant une prime de signature de contrat. La fédération recevant demande le certificat international de transfert(CIT) à son homologue et envoie la copie à la FIFA (Fédération internationale de football) en fin de légaliser le transfert. Après les analyses, la fédération décide d'envoyer ou ne pas envoyer le CIT. Si oui, elle envoie le transfert à la fédération recevant et le transfert est légalisé.

b) Analyse

Dans cette partie nous avons recueillis les données auprès de différents acteurs ayant un rôle au sein de leurs équipes.

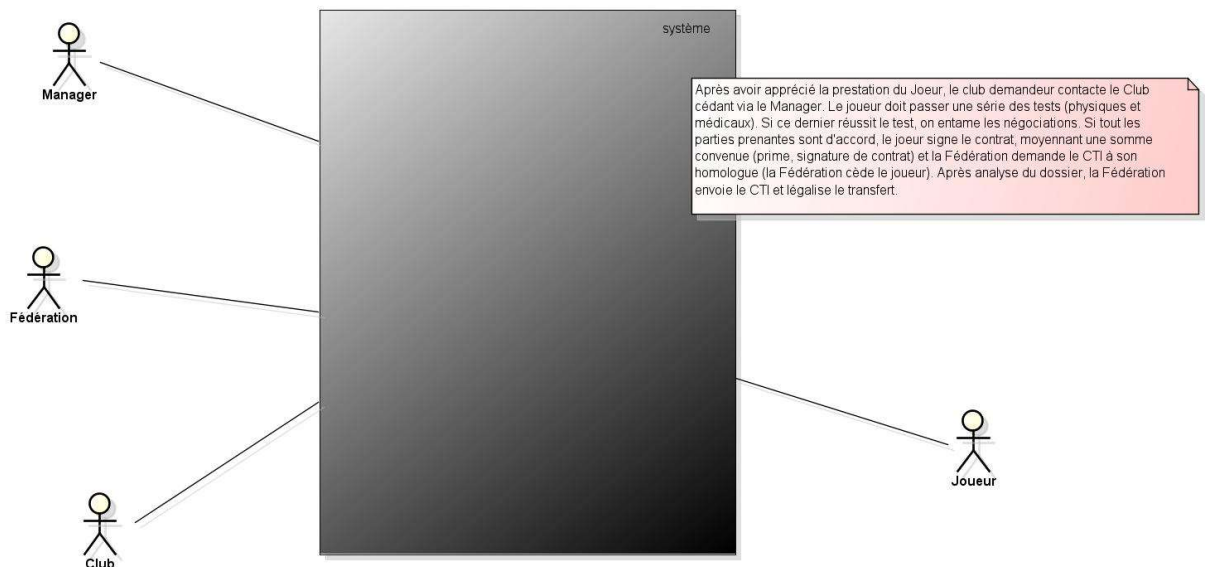


Figure 3 : Diagramme de contexte.

Ce diagramme illustre la relation des acteurs avec le système.

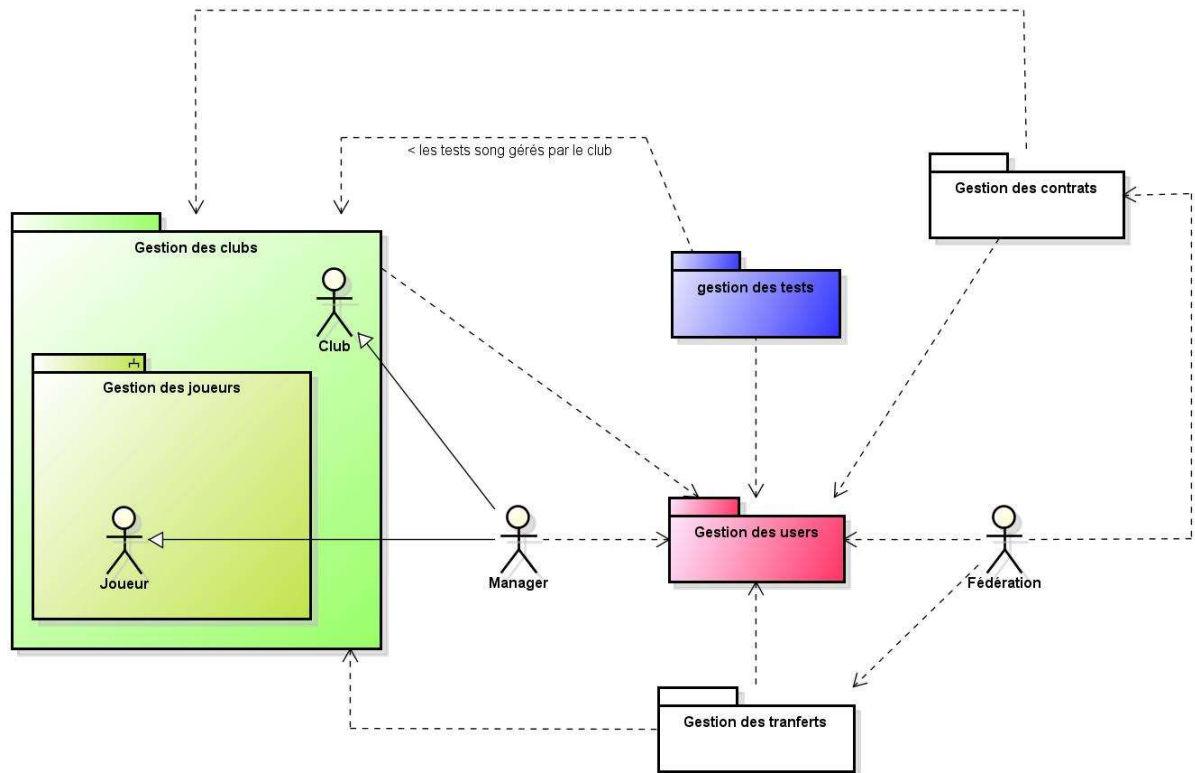


Figure 4 : Diagramme de Package (Paquetage).

Ce diagramme montre les grandes parties de l'application (Subdivision de l'application) ainsi que le rôle ou appartenance de chaque utilisateur par rapport aux fonctionnalités.

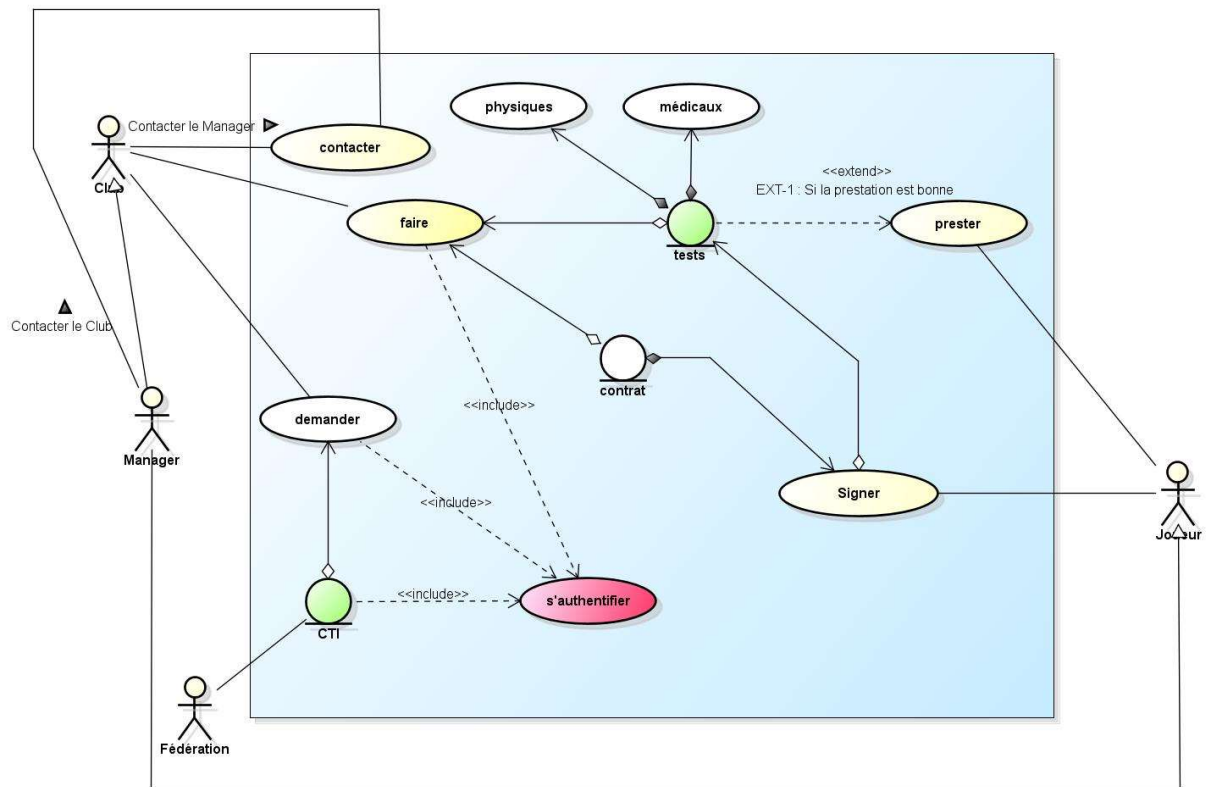


Figure 5 : Diagramme des cas d'utilisation.

Ce diagramme montre l'interaction des acteurs principaux avec le système ainsi que celle du système avec l'acteur secondaire.

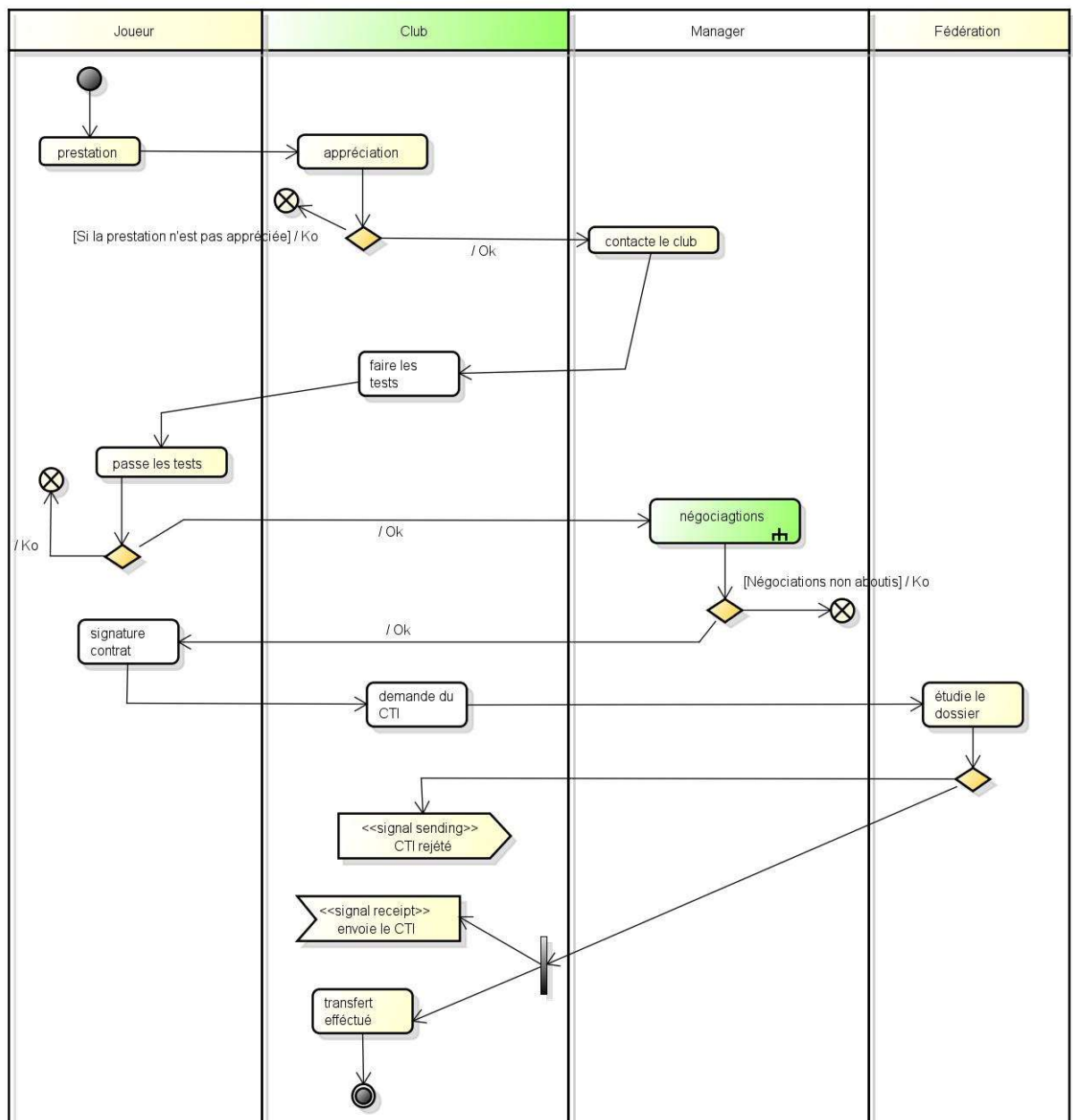


Figure 6 : Diagramme d'activités.

Ici, nous illustrons chaque activité, c'est-à-dire les scénarios possibles.

c) Conception

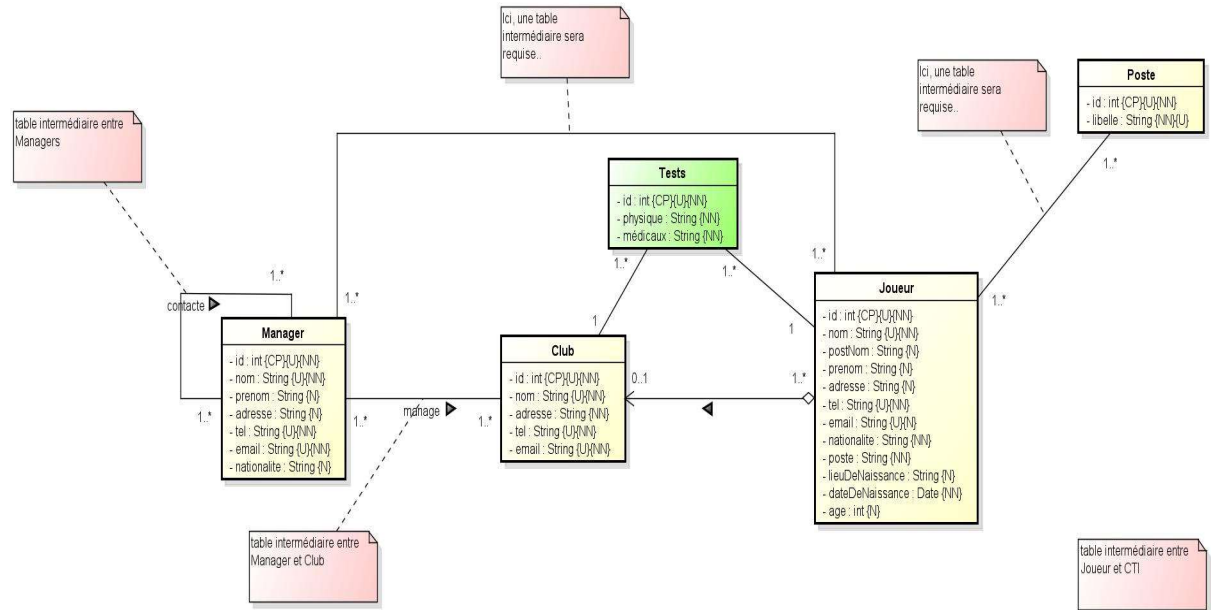


Figure 7 : Diagramme de classe.

Ce diagramme illustre le MCD de la base de données du système.

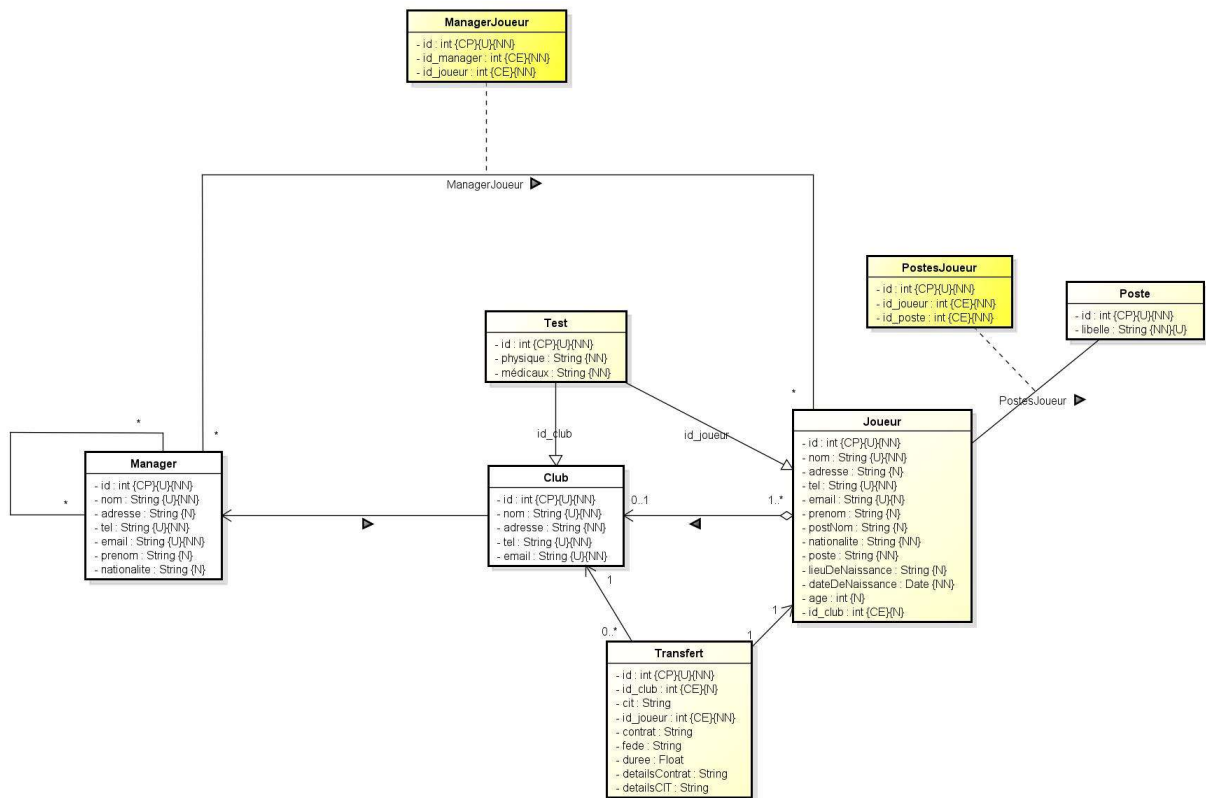


Figure 8 : Diagramme de classe illustrant le MLD de la base de données.

Illustration de la relation et interaction entre entités.

d

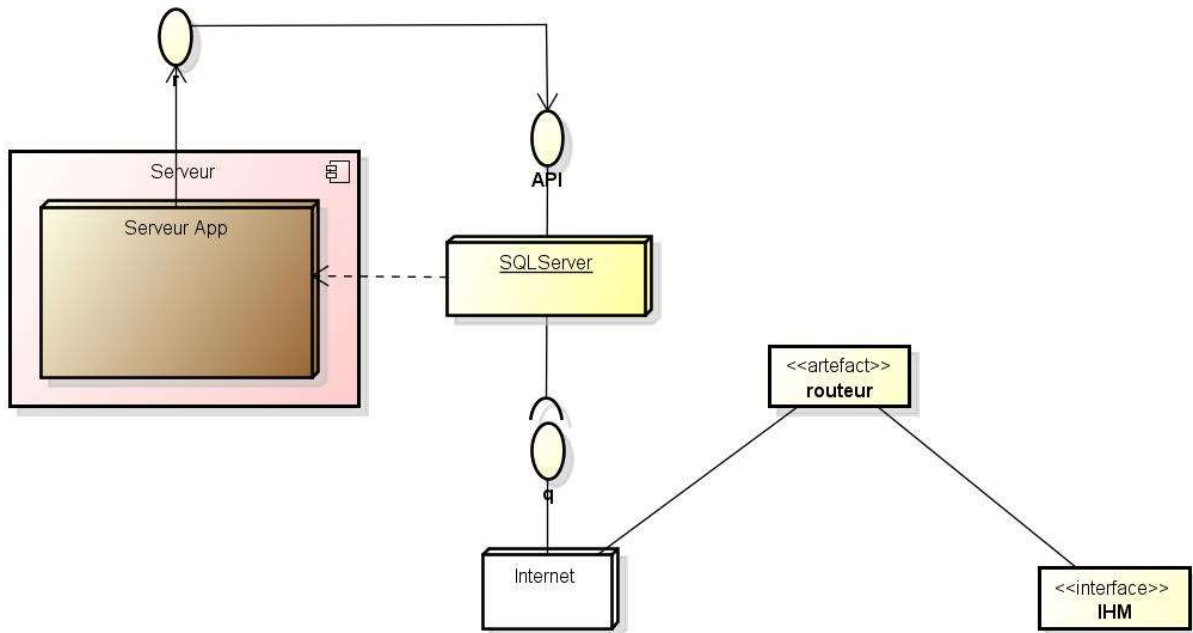


Figure 9 : Diagramme de déploiement.

Ce diagramme montre comment notre système sera déployer. Il illustre également les logiciels et / ou matériel de déploiement.

CHAPITRE III. REALISATION DU SITE WEB

Ce chapitre comprend deux sections à savoir : la création de la base de données et la programmation.

Section 1 : Création de la base de données

Notre préférence s'est porté sur MySQL.

❖ Définition et but

Est un système de gestion de base de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il est développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui des mises à jour fréquentes et fortement sécurisées. Il est *multi-thread* et *multi-utilisateur*.

a) Système d'exploitation supportés

Il fonctionne sur de nombreux système d'exploitation différents, incluant : AIX, IBM i-5, BSDI, FreeBSD, etc.

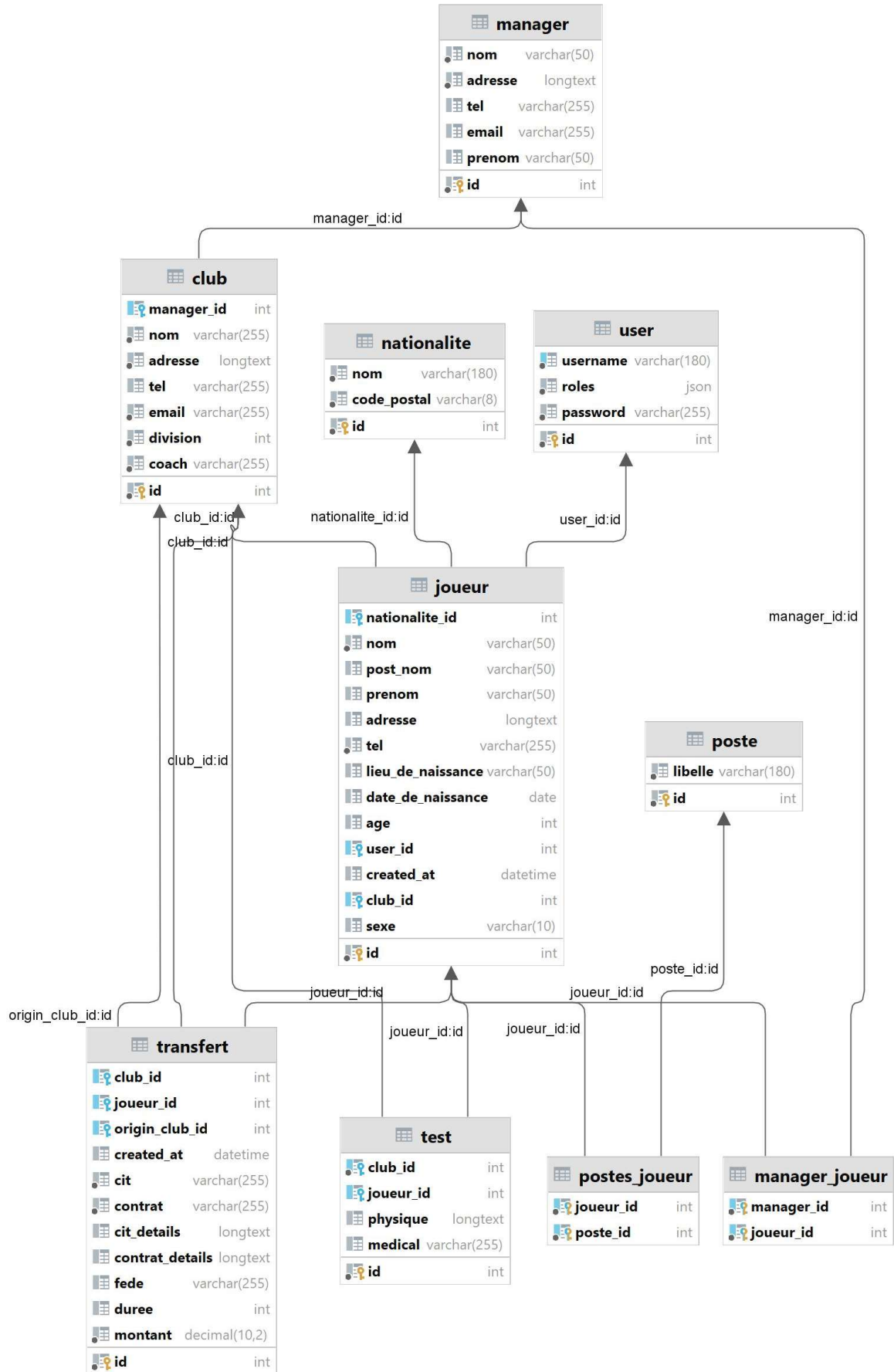
Les bases de données sont accessibles en utilisant les langages de programmation comme : C, C++, VB, VB .Net, C#, Java, PHP, Perl, Python, Windev, Ruby, etc.

b) Langages informatiques supportés

MySQL supporte deux langage informatiques : le langage de requête SQL et le SQL/PSM (Persistent Stored Modules), une extension procédurale standardisée au SQL.

MySQL, permet de combiner des requêtes SQL et des instructions procédurales (boucles, conditions...), dans le but de créer des traitements complexes destinés à être stocké sur le serveur de base de données (objects serveur), par exemple des procédures stockées ou des déclencheurs.

PRESENTATION DE LA BASE DE DONNEES



Section 2 : Programmation

❖ Définition et but

Aussi appelé codage dans le domaine informatique, désigne l'ensemble des activités qui permettent l'écriture des programmes informatiques. C'est une étape importante du développement de logiciels.

Il représente donc, ici, la rédaction du code source d'un logiciel. On utilise plutôt le terme développement pour dénoter l'ensemble des activités à la création d'un logiciel, sa conception, puis son implémentation proprement dite au sens de l'écriture des programmes dans un langage bien défini, ainsi que la vérification de sa correction, etc.

❖ Choix du langage

Notre choix s'est porté sur PHP et JS (JavaScript).

❖ Présentation des interfaces graphiques

The screenshot displays a web application interface. At the top, there is a navigation bar with the logo 'GTJ' on the left and a list of links: 'Accueil', 'À propos', 'News', 'Blog', 'Contact', and 'Connexion'. Below the navigation bar, a breadcrumb trail shows 'Accueil / Connexion', followed by the page title 'Connexion'. The main content area features a centered 'Authentification' form. This form includes a title with a user icon, two input fields labeled 'Username' and 'Mot de passe', and a red 'Connexion' button. At the bottom of the page, there is a dark footer section titled 'Notre Newsletter' with a line of placeholder text and a red 'Souscrire' button.

Tableau de bord

Liste des clubs

Q

#	Désignation	Division	Tél	Email	
#93	OLYMPIC EST	division 2	01 58 05 33 42	martel.paul@laposte.net	
#92	AS RERUM	division 1	+33 (0)4 68 41 85 54	corinne14@orange.fr	
#91	DIGNISSIMOS FC	division 2	02 11 41 38 76	philippine.gerard@gmail.com	
#90	OLYMPIC QUAE	division 3	0276491904	xavier36@adam.fr	
#89	AS DOLORUM	division 3	+33 (0)9 37 67 43 34	gilles.martin@club-internet.fr	
#88	REAL DESERUNT	division 1	0123708146	antoINETte.alves@toussaint.org	

Derniers Transferts

+ Ajouter un joueur | + Ajouter un club

Ajout du club

Formulaire d'ajout

as vclub

contact@vclub.com0904651464

Shungu1

Kinshasa / Limeta

+ Ajouter

CONTACT@EXAMPLE.COM+1 5589 55488 55

Ajout bien effectuée.

AccueilÀ propos

Accueil / Tableau de bord / Ajout du club

Ajout du club

Formulaire d'ajout

* Nom

* Adresse email

N° Tél

* Coach

1

Adresse...

+ Ajouter

Accueil / Tableau de bord / Ajout joueur

Ajout joueur

Formulaire d'ajout

lema

Postnom

AS VCLUB

Prénom

Homme

RÉP. DÉM. DU CONGO

Kinshasa

0900827774

MO MD

05/03/1991

31

Ajouter

Accueil / Tableau de bord / as vclub

as vclub

AS VCLUB

#NationalitéNomÂgePosteTélDernier Club

#17113Rép. Dém. Du CongoLema31 ansMO MD0900827774-

Notre Newsletter

Veuillez renseigner les champs ci-dessous.

Les champs avec la marque "*" sont obligatoires.

Actualisez

9U4EI

01NM8P

Gabonaise

3

23800

ROLAND RUIZ YVES

OLYMPIC EST

AS VCLUB

Description du CIT...

Résumé du contrat...

Valider

[Accueil](#) / [Tableau de bord](#) / [Transferts](#)

Transferts

↔ Transferts

+ Nouveau Transfert

Recherche...

	#	N° CIT	Joueur	Provenance	Destination	Montant [\$]	Durée	Date	
↔	#1	9U4EI	Roland Ruiz Boulanger Yves	↔ OLYMPIC EST	↔ AS VCLUB	23800	3 an(s)	Aujourd'hui à 09:17	

❖ Extrait du code source

```
<?php

namespace App\EventSubscriber;

use ApiPlatform\Core\EventListener\EventPriorities;
use App\Entity\Transfert;
use DateTime;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Component\EventDispatcher\EventSubscriberInterface;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Event\ViewEvent;
use Symfony\Component\HttpFoundation\KernelEvents;

class OnPostTransferEvent implements EventSubscriberInterface
{
    /**
     * @var EntityManagerInterface
     */
    private $em;

    public function __construct(EntityManagerInterface $em)
    {
        $this->em = $em;
    }

    public static function getSubscribedEvents(): array
    {
        return [
            KernelEvents::VIEW => [
                'handler',
                EventPriorities::PRE_VALIDATE,
            ],
        ];
    }

    public function handler(ViewEvent $event)
    {
        $transfer = $event->getControllerResult();
        $method = $event->getRequest()->getMethod();

        if ($transfer instanceof Transfert && $method ===
Request::METHOD_POST) {
            $transfer->setCreatedAt(new DateTime());

            $joueur = $transfer->getJoueur();
            $joueur->setClub($transfer->getClub());

            $this->em->flush();
        }
    }
}
```

■ Code PHP

```
import React, {useEffect, useState} from "react";
import {Col, Row} from "reactstrap";
import {AreaField, InputField} from "../../components/form";
import {handleInputChange, handleItemSelected} from
"../../model/services";
import AsyncSelect from "react-select/async";
import {loadClubs, loadJoueurs} from
"../../model/services/asyncLoadData_Requests";
```

```

import {useDispatch, useSelector} from "react-redux";
import {fetchClubsAction} from "../../model/features/club/clubActions";
import {fetchJoueursAction} from
"../../model/features/joueurs/joueursActions";
import {Spinner} from "react-bootstrap";
import {useNavigate, useParams} from "react-router-dom";
import {toast} from "react-toastify";
import {onInitTransferMessage} from
"../../model/features/transfers/transferSlice";
import {
  editTransferAction,
  findTransferAction,
  postTransferAction
} from "../../model/features/transfers/transferActions";
import {onRefreshClubs} from "../../model/features/club/clubSlice";
import {onRefreshJoueurs} from "../../model/features/joueurs/joueursSlice";

export const TransferForm = () => {

  const dispatch = useDispatch()
  const navigate = useNavigate()

  const { id } = useParams()

  const [ dJoueurs, setDJoueurs ] = useState([ ])
  const [ dClubs, setDClubs ] = useState([ ])
  const [ dOriginClubs, setDOriginClubs ] = useState([ ])
  const [ joueur, setJoueur ] = useState(null)
  const [ club, setClub ] = useState(null)
  const [ originClub, setOriginClub ] = useState(null)
  const [ transfer, setTransfer ] = useState({
    cit: '', contrat: '', citDetails: '', contratDetails: '', fede: '',
    duree: 1,
    montant: 0,
  })

  const { clubs, status, loader } = useSelector(state => state.club)
  const { message, cLoader, eLoader, transfer: data } = useSelector(state
=> state.transfers)
  const { joueurs, status: jStatus, loader: jLoader } = useSelector(state
=> state.joueurs)

  useEffect(() => {
    if (status === 'idle') dispatch(fetchClubsAction())
    if (jStatus === 'idle') dispatch(fetchJoueursAction())
  }, [dispatch, status, jStatus])

  useEffect(() => {
    if (clubs && clubs.length > 0) {
      const obj = clubs.map(club => {
        return {
          id: club.id,
          label: club.nom.toUpperCase(),
          value: `/api/clubs/${club.id}`,
          data: club,
        }
      })
      setDClubs(obj)
      setDOriginClubs(obj)
    }
  }, [clubs])

  useEffect(() => {
    if (joueurs && joueurs.length > 0) {

```

```

    setDJoueurs(joueurs.map(joueur => {
      const prenom = joueur.prenom ? joueur.prenom.toUpperCase() : ''
      return {
        id: joueur.id,
        label: joueur.nom.toUpperCase() + ' ' + prenom,
        value: `/api/joueurs/${joueur.id}`,
        data: joueur,
      }
    }
  ))
}, [joueurs])

useEffect(() => {
  if (message) {
    toast.success(message)
    dispatch(onInitTransferMessage())
    navigate('/transfers')
  }
}, [dispatch, navigate, message])

useEffect(() => {
  if (data) {
    setTransfer({
      id,
      cit: data.cit,
      contrat: data.contrat,
      fede: data.fede,
      duree: data.duree,
      montant: parseFloat(data.montant),
      citDetails: data.citDetails ? data.citDetails : '',
      contratDetails: data.contratDetails ? data.contratDetails : '',
    })

    const prenom = data.joueur.prenom ? data.joueur.prenom : ''
    setJoueur({
      label: data.joueur.nom.toUpperCase() + ' ' + prenom.toUpperCase(),
      value: `/api/joueurs/${data.joueur.id}`,
    })

    setClub({
      label: data.club.nom.toUpperCase(),
      value: `/api/clubs/${data.club.id}`,
    })

    setOriginClub({
      label: data.originClub.nom.toUpperCase(),
      value: `/api/clubs/${data.originClub.id}`,
    })
  }
}, [data, id])

useEffect(() => {
  if (id && !transfer) dispatch(findTransferAction(id))
}, [transfer, dispatch, id])

function handleRefresh() {
  dispatch(onRefreshClubs())
  dispatch(onRefreshJoueurs())
}

function handleJoueurChange(event) {
  if (event) {
    setOriginClub({
      label: event.data.club.nom.toUpperCase(),

```

```

        value: `/api/clubs/${event.data.club.id}`,
    })
    setJoueur(event)
} else {
    setOriginClub(null)
    setJoueur(null)
}
}

function handleReset() {
    setJoueur(null)
    setClub(null)
    setOriginClub(null)
    setTransfer({
        cit: '',
        contratDetails: '',
        fede: '',
        citDetails: '',
        contrat: '',
        duree: 1,
        montant: 0,
    })
}

function onSubmit(e) {
    e.preventDefault()
    if (id) {
        dispatch(editTransferAction({
            ...transfer,
            duree: transfer.duree ? parseInt(transfer.duree) : null,
            montant: transfer.montant || !isNaN(transfer.montant) ?
transfer.montant.toString() : '0',
            club: club ? club.value : null,
            originClub: originClub ? originClub.value : null,
            joueur: joueur ? joueur.value : null,
        }))
    } else {
        dispatch(postTransferAction({
            ...transfer,
            duree: transfer.duree ? parseInt(transfer.duree) : null,
            montant: transfer.montant || !isNaN(transfer.montant) ?
transfer.montant.toString() : '0',
            club: club ? club.value : null,
            originClub: originClub ? originClub.value : null,
            joueur: joueur ? joueur.value : null,
        }))
    }
}

return (
    <form onSubmit={onSubmit} className='php-email-form'>
        <p>
            <button type='button' className="btn btn-primary" disabled={jLoader
|| cLoader || loader} onClick={handleRefresh}>
                {(jLoader || cLoader) && 'Actualisez'}
                {(jLoader || cLoader) && <><Spinner animation='grow' size='sm' />
Chargement en cours.</>}
            </button>
        </p>
        <Row>
            <Col lg={7} md={6}>
                <InputField
                    isRequired
                    isDisabled={loader || eLoader || cLoader || jLoader}

```

```

        onChange={ (e) => handleInputFieldChange(e, transfer,
setTransfer)}
        value={transfer.cit}
        name='cit'
        className='mb-3'
        placeholder='* N° CIT' />
<InputField
  isRequired
  isDisabled={loader || eLoader || cLoader || jLoader}
  onChange={ (e) => handleInputFieldChange(e, transfer,
setTransfer)}
  value={transfer.contrat}
  name='contrat'
  className='mb-3'
  placeholder='* Réf. Contrat' />
<InputField
  isRequired
  isDisabled={loader || eLoader || cLoader || jLoader}
  onChange={ (e) => handleInputFieldChange(e, transfer,
setTransfer)}
  value={transfer.fede}
  name='fede'
  className='mb-3'
  placeholder='* Fédération' />
<InputField
  isRequired
  type='number'
  isDisabled={loader || eLoader || cLoader || jLoader}
  onChange={ (e) => handleInputFieldChange(e, transfer,
setTransfer)}
  value={transfer.duree}
  name='duree'
  className='mb-3'
  placeholder='* Durée du contrat' />
<InputField
  isRequired
  type='number'
  isDisabled={loader || eLoader || cLoader || jLoader}
  onChange={ (e) => handleInputFieldChange(e, transfer,
setTransfer)}
  value={transfer.montant}
  name='montant'
  className='mb-3'
  placeholder='* Montant ($)' />

<div className="mb-3">
  <AsyncSelect
    isClearable
    cacheOptions
    isDisabled={loader || eLoader || cLoader || jLoader}
    placeholder='* Joueur'
    value={joueur}
    onChange={ (e) => handleJoueurChange(e) }
    defaultOptions={dJoueurs}
    loadOptions={loadJoueurs} />
</div>

<div className="mb-3">
  <AsyncSelect
    isClearable
    cacheOptions
    isDisabled={loader || eLoader || cLoader || jLoader}
    placeholder='* Club de Provenance'
    value={originClub}

```

```

        onChange={ (e) => handleItemSelected(e, setOriginClub)}
        defaultOptions={dOriginClubs}
        loadOptions={loadClubs} />
    </div>

    <div className="mb-3">
        <AsyncSelect
            isClearable
            cacheOptions
            isDisabled={loader || eLoader || cLoader || jLoader}
            placeholder='* Club de Destination'
            value={club}
            onChange={ (e) => handleItemSelected(e, setClub)}
            defaultOptions={dClubs}
            loadOptions={loadClubs} />
    </div>
</Col>

<Col lg={5} md={6}>
    <AreaField
        isDisabled={loader || eLoader || cLoader || jLoader}
        value={transfer.citDetails}
        name='citDetails'
        onChange={ (e) => handleInputChange(e, transfer,
setTransfer)}
        className='mb-3'
        placeholder='Description du CIT...' />
    <AreaField
        isDisabled={loader || eLoader || cLoader || jLoader}
        value={transfer.contratDetails}
        name='contratDetails'
        onChange={ (e) => handleInputChange(e, transfer,
setTransfer)}
        className='mb-3'
        placeholder='Résumé du contrat...' />
    </Col>
</Row>

<div className="text-end">
    {!id && (
        <button
            type='button'
            className="btn border-0"
            disabled={loader || eLoader || cLoader || jLoader}
            onClick={handleReset}>
            <i className='far fa-trash-alt' />
        </button>
    )}

    <button type='submit' className="btn border-0" disabled={loader ||
cLoader || jLoader}>
        {(cLoader || eLoader) && !id && 'Valider'}
        {(cLoader || eLoader) && id && 'Modifier'}
        {(cLoader || eLoader) && <><Spinner animation='grow' size='sm' />
        Veuillez patienter.</>}
    </button>
</div>
</form>
)
}

```

CONCLUSION.

Nous voici au terme de notre travail qui a porté sur la « **Gestion des Transferts des Joueurs** » cas de la RD Congo.

Nos investigations nous ont emmenés à nous demander si les équipes de football en RDC ont les moyens favorables, pour gérer les transferts des joueurs.

Pour atteindre ce but nous nous sommes fixé comme objectif de doter un site web dynamique au club, enfin d'améliorer et promouvoir d'avantage la gestion des transferts de joueurs.

Pour ce faire, nous avons subdivisé notre travail en trois chapitres outre l'introduction et la conclusion.

Le premier chapitre a porté sur les **CONCEPTS THEORIQUES DE BASE**. Il s'agit dans ce chapitre d'expliquer et d'assimiler la / les théorie(s) liées aux concepts cadrant le Web et aux bases de données, qui constituent l'épine dorsale de la gestion informatique et à la bonne compréhension de ce travail. Le deuxième Chapitre a porté sur la **CONCEPTION DU SITE**. Il s'agit dans ce chapitre du développement de la procédure ou méthodologie enfin d'aboutir à un résultat escompté et pour ce faire, nous avons utilisé **UML** comme langage et méthode de modélisation. Outre ces étapes, nous avons pu mettre un accent sur la critique de la gestion du transfert des joueurs incluant les clubs, les joueurs, les fédérations, etc. cette présentation nous a été d'une grande utilité parce qu'elle nous permet de connaître l'impact de cette solution (notre étude) sur les différentes parties prenantes.

Et enfin le troisième chapitre a porté sur la **REALISATION DU SITE**. Dans ce dernier, nous avons mis en avant la solution, en intégrant quelques états de sorties. Il est d'une grande importance car il nous permet de connaître à la fois l'application mais aussi les outils qui nous ont permis de réaliser le site, entre autre :

- ✓ Laragon : Est un paquetage Serveur qui intègre Apache, PHP ainsi que MySQL ;
- ✓ Apache : Serveur d'application orienté Web ;
- ✓ PHP : Un langage de programmation (POO) orienté serveur ;
- ✓ JavaScript (JS) : langage de programmation événementiel ;
- ✓ MySQL : Un système de gestion de base de données.

Nous remarquons que la technologie évolue au jour le jour et prends de plus en plus du terrain dans notre Pays. Avec les dernières avancées de pointes dans le milieu du Sport et du Football en particulier, nous trouvons qu'il y a un écart entre la réalité et l'évolution ; d'où la nécessité de doter nos équipes (clubs) ainsi que notre fédération des outils adéquat enfin de faciliter la gestion de données ainsi que leur tâche du quotidien.

BIBLIOGRAPHI

Notes des cours

PONPON NGAVU »Cours inédit de méthodologie de recherche scientifique »ULK/Limete, Kinshasa, 2019-2020.

Prof ISAKATONGA LOANIE JUSTIN »Cours de technique de base de données », ULK/Limete, Kinshasa 2020-2021.

INERNET.

Cours d'uml

Mercato

Les diagrammes

Informations sur le site web.