## RESUME

L’objectif de ce projet est de mettre à la disposition du centre médical interentreprises(CMI) de Orange cote d’Ivoire Telecom(OCIT) une nouvelle application de gestion médicale. Cette nouvelle application prend en compte les besoins du centre qu’ils soient de la gestion des activités du circuit curatif, des activités du circuit préventif, de la gestion du stock pharmaceutique et la gestion des statistiques.

L’étude globale de la solution existante nous a permis de proposer des solutions appropriées. Nous avons procédé dans un premier temps à la formalisation des besoins du CMI qui s’est aboutie à la modélisation de ceux-ci. Ensuite nous avons procédés à la réalisation du projet en s’inspirant des modèles. Enfin des tests ont été réalisés sur la nouvelle solution avant la mise en production et la formation des Agents du CMI.

Soulignons que cette application permettra à Orange côte Télécom de suivre en temps Réel l’état de santé de ses agents, et quant au CMI elle lui permettra de gérer aisément le traitement d’un patient.

## INTRODUCTION

L’avènement des technologies de l’information et de la communication avec les services qu’elles proposent, a révolutionné les habitudes des hommes ainsi que des entreprises. Actuellement, l’informatique est au cœur de toutes entreprises. De ce fait, loin d’être un effet de mode, l’informatisation des systèmes devient non seulement nécessaire mais capitale.

Dès lors, Leader de la téléphonie en Côte d’Ivoire, Orange Côte d’ivoire Télécom (OCIT) a entrepris depuis quelques années un vaste chantier de transformation de son système d’informations pour le rendre plus concurrentiel et plus stable. Cette transformation s’est accompagnée d’une ouverture de plus en plus croissante vers la digitalisation dans plusieurs de ses secteurs d’activités mais aussi dans celui du secteur médical. Cela s’est traduit depuis quelques années à la mise en place d’une plateforme de gestion des dossiers médicaux de messages des agents (PRANCME). Cependant cette plateforme est devenue obsolète et n’est plus maintenue.

Ainsi, il en est conclu la nécessité d’une nouvelle plateforme pour la gestion des dossiers médicaux des agents remédiant aux difficultés identifiées dans l’application existante. Pour cela Orange Côte d’Ivoire Telecom a sollicité Smile CI pour la mise en place de cette plateforme afin de corriger les imperfections et les insuffisantes de l’existant et surtout de prendre en compte les nouvelles évolutions du système. Et C’est dans ce cadre qu’elle nous a confié le projet CMI dont le thème est **la mise en place d’une application de gestion d’un centre médical interentreprises**.

Cette nouvelle plateforme devra permettre à Orange cote d’ivoire Telecom de suivre l’état de santé des agents et de leurs ayants droit d’Orange Cote d’ivoire télécom.

Dans la suite, nous articulons notre autour de trois grandes parties : **l’étude** **préalable**  dans laquelle nous présenterons d’abord le contexte dans lequel s’inscrit le projet et ensuite le thème qui nous a été soumis,  **l’étude conceptuelle**  où nous présenterons la méthode d’analyse ainsi que les différentes modélisations du futur système et enfin **une étude technique** qui aboutira à la réalisation de notre solution.

**PARTIE**



Dans cette partie, nous présenterons succinctement le contexte dans lequel se situe le projet, nous décrierons par la suite l’existant, apporter des critiques sur l’existant tout en proposant des solutions. Elle consistera également en la présentation de la structure d’accueil ainsi que du cahier de charge.

# 

# ETUDE TECHNIQUE ET REALISATION

## CHAPITRE V- ETUDE TECHNIQUE

Dans l’étude technique allons procéder à des études comparatives afin de déterminer les contraintes et spécifications technique liées au système. Il s’agit entre autre du choix du système de gestion de la base de données, d’un serveur d’application, de l’architecture de déploiement de notre solution, et enfin des langages de programmation.

### V.1- Choix technique

#### V.1.1- Choix du SGBD

La fonction première d'un Système de Gestion de Base de Données (SGBD) est d'être un outil de stockage d'informations offrant des fonctions de manipulation des données. L’un des avantages des SGBD est que l'interrogation de ces informations s’effectue d’une manière indépendante de l'architecture physique de stockage. Il existe beaucoup de SGBD cependant nous allons étudier les plus populaires dans et les plus accessibles pour sélectionner le plus adapté à notre projet.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NOM DU SGBD | SUPPORT DES TRANSACTIONS | RAPIDITE | GRATUITE | GRANDE RESISTANCE | SIMPLICITE  D’ADMINISTRATION | MULTI PLATEFORME |
| ORACLE |  |  |  |  |  |  |
| SQL SERVER |  |  |  |  |  |  |
| MYSQL/  MARIADB |  |  |  |  |  |  |

*Tableau 15 : Comparaison des SGBD*

Au vu de ce comparatif, on note que MARIADB a une prise en main simple et facile et est gratuit. De plus, SMILE CI est une entreprise spécialisé dans la proposition open source. Ce qui nous permet d’orienter notre choix vers MARIADB.

#### V.1.2- Choix des langages de programmation

* **Langage de scripts serveur**

En quelques années seulement, la conception des sites web a radicalement changé. Les sites doivent être de plus en plus riches en informations ce qui oblige à des mises à jour automatisées. C'est l'une des fonctions d'un langage de script côté serveur comme la connexion à une base de données, la génération d'outils de recherche pour les visiteurs, la personnalisation des pages.

* **ASP.NET**

ASP.NET est un ensemble de technologies de programmation web créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites web dynamiques, des applications web ou des web services XML. La technologie est accessible grâce à l'installation d'un serveur web compatible ASP (IIS) ou à l'intérieur de Visual Web Developer Express Edition. ASP.NET fait partie de la plateforme Microsoft .NET et est le successeur de la technologie Active Server Pages (ASP).

* **PHP**

PHP (HyperText Preprocessor), est un langage de scripts libres principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande.

* **JEE**

**Java Enterprise Edition**, ou **Java EE** (anciennement **J2EE**), est une [spécification](https://fr.wikipedia.org/wiki/Sp%C3%A9cification_(norme_technique)) pour la [technique](https://fr.wikipedia.org/wiki/Technique) [Java](https://fr.wikipedia.org/wiki/Java_(technique)) d'[Oracle](https://fr.wikipedia.org/wiki/Oracle_(entreprise)) plus particulièrement destinée aux applications d’entreprise. Ces applications sont considérées dans une approche [multi-niveaux](https://fr.wikipedia.org/wiki/Client-serveur#Architecture_.C3.A0_3_niveaux). Dans ce but, toutes les implémentations de cette spécification contiennent un ensemble d’extensions au [*framework*](https://fr.wikipedia.org/wiki/Framework) Java standard ([JSE](https://fr.wikipedia.org/wiki/J2SE), *Java Standard Edition*) afin de faciliter notamment la création d’[applications réparties](https://fr.wikipedia.org/wiki/Architecture_distribu%C3%A9e).

* **Choix du langage**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LANGAGE | MULTIPLATEFORME | SÉCURITÉ | FORTE  COMMUNAUTÉ | CMPLEXITE | PUISSANCE |
| Asp |  |  |  |  |  |
| PHP |  |  |  |  |  |
| JEE |  |  |  |  |  |

Tableau 19: Comparaison des langages de scripts

* **Langages coté clients**

Les langages coté client sont essentiellement du Html, css et le *framework AngularJs* qui se base sur du javascrit pour dynamiser le Html. Le framework est adapté à la création des Application single page(SPA).

#### V.1.1- Choix du serveur d’application (serveurs web)

Aujourd'hui, la principale mission des serveurs web porte sur l'accès aux applications et aux données de l'entreprise. Et ce, au moyen de transactions effectuées par des interpréteurs de scripts ou des classes techniques (Perl, VB, JS, ADO.NET, JDBC...) qui traduisent des requêtes HTTP en commandes adressées au système d'exploitation.

L'autre type de transaction repose sur l'accès à un site sécurisé. Les requêtes HTTP sont alors chiffrées sur une couche SSL (HTTPS) afin d'accéder à un serveur HTTP qu'authentifie un certificat.Plusieurs années d’existence ont consolidé le marché des serveurs au point qu’il ne regroupe aujourd’hui qu’une poignée de produits. Nous pouvons citer entre autres Apache et Microsoft IIS

* Apache

Apache est apparu en avril 1995. Sa nouvelle version (version 2) possède plusieurs avancées majeures, entre autres :

* le support de plusieurs plateformes,
* le support de processus légers UNIX,
* une nouvelle API et le support IPv6.

Il est à noter que le serveur Apache fonctionne principalement sur les systèmes d'exploitation Unix (GNU/Linux, BSD et UNIX) et Windows. La version Windows n'est considérée comme stable que depuis la version 2 d'Apache. Coté conception, Apache est conçu pour supporter de nombreux modules lui donnant des fonctionnalités supplémentaires :

* interprétation du langage Perl, PHP et Python,
* serveur proxy,
* Common Gateway Interface,
* Server SideIncludes,
* réécriture d'URL, négociation de contenu,
* protocoles de communication additionnels,
* Microsoft IIS

Microsoft IIS (Internet Informations Services) est le serveur HTTP créé par Microsoft. Ce serveur Web n'est utilisable que sur des produits de Microsoft (Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Server 2005 et Windows server 2008).

Au début de l'histoire de ce produit, il y a eu de nombreux problèmes de sécurité. La mise en place de site était très abordable mais il n'y avait aucune sécurité. Microsoft est donc repartie de zéro pour la sixième version de leur serveur.

Au volet fonctionnalités, Microsoft IIS :

* supporte de nombreuses technologies telles que ASP, .NET, le PHP ou encore le CGI.
* utilise une métabase texte qui permet de faire des sauvegardes et des restaurations en cas de problèmes critiques sur le serveur.
* possède des fonctionnalités de résolutions de problèmes et la possibilité de récupération de métabase endommagée.
* détecte également les problèmes de pertes de mémoire, les violations d'accès et autres erreurs.
* permet la tolérance des pannes et le redémarrage des processus si nécessaire.
* ***Choix du serveur web***

Afin de faire un choix judicieux, nous allons établir un tableau de comparaison des serveurs web dominants du marché.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom du serveur | Prise en main et installation | Gratuité | Sécurité | Multiplateforme |
| Apache | **validN2.png** | **validO2.png** | **validO2.png** | **validO2.png** |
| Microsoft IIS | **validO2.png** | **validN2.png** | **validO2.png** | **validN2.png** |

Tableau 20 : étude comparative de serveur d’application

Après analyse du tableau de comparaison, nous optons pour le serveur web Apache car il offre une bonne sécurité, il est gratuit et multiplateforme;

### V.2- Architecture matérielle et logicielle

#### V.2.1- Description de l’architecture

En règle générale, une application informatique peut être découpée en trois niveaux d'abstraction distincts :

* **La couche de présentation**, encore appelée interface homme-machine (IHM), permet l’interaction de l’application avec l’utilisateur. Cette couche gère les saisies au clavier, à la souris, et à la présentation des informations à l’écran. Dans la mesure du possible, elle doit être conviviale et ergonomique ;
* **La logique applicative**, les traitements, décrivant les travaux à réaliser par l’application. Ils 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 à la saisie,
* les traitements globaux, constituant l’application elle-même. Cette couche appelée Business Logic ou couche métier, contient les règles internes qui régissent une entreprise donnée.
* **Les données**, encore plus exactement l'accès aux données, regroupant l'ensemble des mécanismes permettant la gestion des informations stockées par l'application.

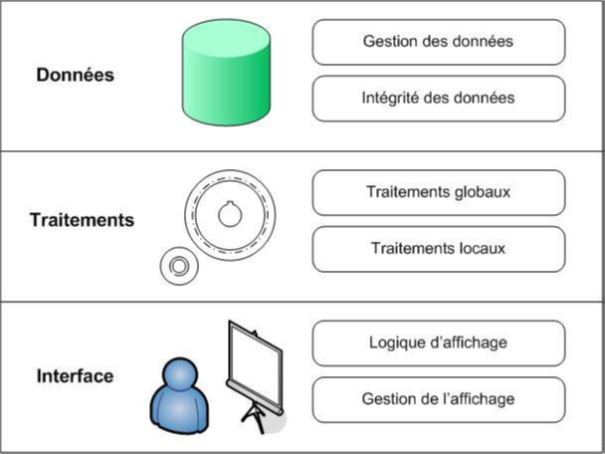


Figure 29 : les différentes couches d’une application web

Ces trois niveaux pouvant être imbriqués ou repartis de différentes manières entre plusieurs machines, leur découpage et leur répartition permettent de distinguer plusieurs architectures applicatives.

* **Architecture un tiers**

Dans une application un tiers, les trois couches applicatives sont intimement liées et s’exécutent sur le même ordinateur. Nous ne parlons pas ici d’architecture client-serveur, mais d’informatique centralisée.

Les utilisateurs se connectent aux applications exécutées par le serveur central (le mainframe) à l’aide de terminaux passifs se comportant en esclaves. C’est le serveur central qui prend en charge l’intégralité de traitements, y compris l’affichage qui est simplement déporté sur des terminaux passifs.

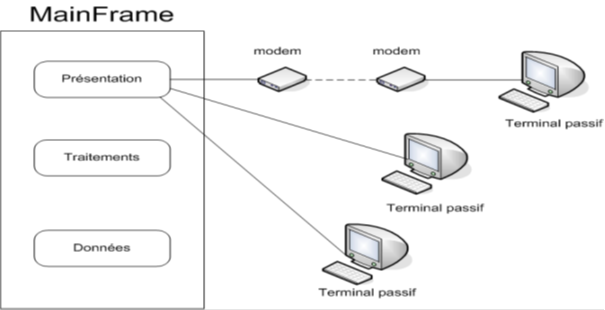


Figure 30:Architecture d’une application sur site central

* **Architecture deux tiers**

Dans une architecture deux tiers, encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé. Ce type d'application permet de tirer parti de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui sont gérées de façon centralisée.

Le modèle client-serveur met en œuvre une conversation entre deux programmes que l'on peut opposer à l'échange figé « maître-esclave » qu'entretiennent les applications sur site central avec leurs terminaux passifs. Lors d’une telle conversation, on distingue les deux parties suivantes :

* **Le client**, c'est le programme qui provoque le dialogue,
* **Le serveur**, c'est le programme qui se contente de répondre au client.

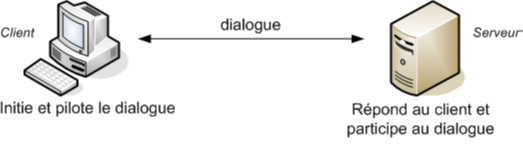


Figure 31:Dialogue client-serveur

* Architecture trois tiers

L’architecture trois tiers applique les principes suivants :

* les données sont toujours gérées de façon centralisée
* la présentation est toujours prise en charge par le poste client
* la logique applicative est prise en charge par un serveur intermédiaire

Par conséquent, cette architecture appelée encore client-serveur de deuxième génération ou client-serveur distribué, sépare l’application en trois niveaux de service distincts :

* **premier niveau** : l’affichage et les traitements locaux (contrôles de saisie, mise en forme de données…) sont pris en charge par le poste client,
* **deuxième niveau** : les traitements applicatifs globaux sont pris en charge par le service applicatif,
* **troisième niveau :** les services de base de données sont pris en charge par un système de gestion de base de données (SGBD).

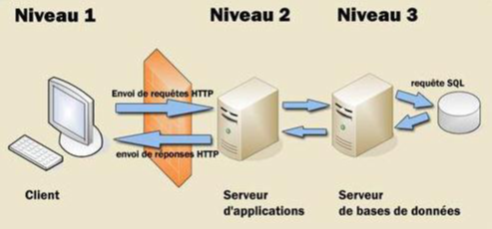


Figure 32:Architecture 3-tiers

Dans le cadre d’une application web, le poste client prend la forme d’un navigateur, le service applicatif est assuré par un serveur HTTP. Nous avons aussi, une nette distinction entre deux tronçons de communication indépendants et délimités par le serveur HTTP.

* Le premier tronçon relie le poste client au serveur Web pour permettre l'interaction avec l'utilisateur et la visualisation des résultats. On l'appelle circuit froid et n'est composé que de standards (principalement HTML et HTTP). Le serveur web tient le rôle de « façade HTTP » ;
* Le deuxième tronçon permet la collecte des données, il est aussi appelé circuit chaud. Les mécanismes utilisés sont comparables à ceux mis en œuvre pour une application deux tiers. Ils ne franchissent jamais la façade http, et de ce fait, peuvent évoluer sans impacter la configuration des postes clients.

#### V.2.2- Choix de l’architecture

Afin de choisir une architecture pour l’application, nous avons procédé à une étude comparée des différentes architectures précédemment présentées.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Architecture | Sécurité | Optimisation des ressources | Richesse de l’interface utilisateur | Simplicité d’administration |
| Un tiers | **validN2.png** | **validO2.png** | **validN2.png** | **validO2.png** |
| Deux tiers | **validO2.png** | **validN2.png** | **validO2.png** | **validN2.png** |
| Trois tiers | ***validO2.png*** | ***validO2.png*** | ***validO2.png*** | ***validN2.png*** |

Tableau 21:Comparaison des architectures d’applications

Au vu de ce tableau, et compte tenu du cadre professionnel et institutionnel du projet, notre choix s’est porté sur une architecture trois tiers puisqu’elle offre une grande marge d’évolution et est assez flexible.

## CONCLUSION

Durant ce stage effectué à Smille Côte d’Ivoire, il nous a été demandé de mettre en place « une application de gestion d’un centre Médical interentreprises » suite à la réponse à l’appel d’offre d’orange côte et Telecom. Cette application qui modélise le processus de gestion d’un patient depuis l’accueil, aux diagnostics, aux traitements jusqu’à la prescription des ordonnances, a pour objectif est non seulement un outil d’aide aux agents de CMI mais permet également de suivre l’état de santé des agents d’orange Cote d’Ivoire et Télécom.

Partant de la prise de l’analyse de l’application existante nous avons proposé la création d’une application de gestion du centre Médical CMI prenant en compte les exigences du cahier des charges qui nous a été soumis et palliant aux insuffisances de la première. Il s’agit pour nous principalement d’implémenter le module accueil, archiviste, infirmier, Médecin, pharmacien et administration. Pour la réalisation nous avons procédé à une étude du système grâce aux processus unifié qui a permis d’aboutir à la conception et la modélisation des objets du système.

Le travail qui nous a été confié est à ce jour réalisé en partie. La première version est en production.

En ce qui nous concerne, nous sommes convaincus que cette application permettra d’une part d’augmenter la rentabilité du CMI en diminuant le temps de traitement au grand bénéfice des patients et d’autre part permettre à la direction de Orange et Cote d’Ivoire Telecom d’avoir une vision globale du fonctionnement du centre Médical. Par ailleurs ce stage nous a permis de nous frotter au milieu de l’entreprise, de conduire un projet de bout en bout. C’est aussi moment de signaler nous avons approfondis nous dans le domaine de la programmation mobile (Ionic, ios, Windows phone, Android), le traitement des données full texte avec Elasticsearch, la gestion des processus métier avec W4, les ERP tel que Odoo Drupal et Magento.