DEDICACE

A Mon Père

Merci pour tout

REMERCIEMENT

La réalisation de ce rapport s’est faite grâce au soutien de l’**Eternel.**

Au terme de ce travail, nous tenons à exprimer notre reconnaissance et notre profonde gratitude à :

* *Dr.* ***OUAFO Blaise,*** *Directeur de IME, pour la formation de qualité qu’il met à notre disposition ;*
* *M.* ***KENFACK Bienvenu,*** *notre encadreur académique, pour sa disponibilité, son soutien et son attention à notre égard ;*
* *M.* ***FOKENG Silva,*** *responsable du département recherche et développement, qui nous a permis d’effectuer un stage chez SOFINEZ pour son soutien, pour le temps qu’il nous a consacrés, ainsi que les conseils utiles qu’il nous a donnés tout au long de notre stage.*
* *Nos vifs remerciements s’adressent également aux membres du jury qui ont accepté d’évaluer notre travail*
* *Tous les enseignants d’IME à qui nous devons du respect. Et le personnel administratif pour leur sympathie et aux services qu’ils nous ont rendus ;*
* ***Mon père****,* ***maman Germaine*** *pour votre patience, vos sacrifices et votre soutien tout au long de mon parcours ;*
* *A l’ensemble de ma famille, à mes grands-parents pour leur accompagnement tout au long de ma vie ;*
* *A mes chers frères, sœurs, cousins et cousines avec qui j’ai passés des instants inoubliables. Je ne saurais exprimer ma reconnaissance la plus sincère et mon respect le plus profond pour tout votre amour, ainsi qu’à vos encouragements qui ont été pour moi un grand soutien moral tout au long de mes études ;*
* *A mes camarades de promotions ;*

A tous ceux qui m’ont soutenu, **Merci** infiniment.

PRESENTATION DE L’ENTREPRISE

1. ENVIRONNEMENT INTERNE DE L’ENTREPRISE
2. HISTORIQUE

Créée en 1999, par un groupe d’ingénieurs polytechniciens, Sécurité Electronique (SECEL) est une SARL camerounaise dont la vocation première est d’être à l’écoute de ses clients et leur apporter la solution idoine à leurs problèmes liés aux Technologies de l’Information et de la Communication (TIC). Au cours des années, cette vocation s’est consolidée. C’est ainsi que la fourniture des solutions matérielles, logicielles et managériales, a connu un fort développement. Toujours à cette période, la fourniture des services comme la maintenance informatique, l’infogérance, et la formation n’ont pas été en reste. Aujourd’hui, SECEL continue son expansion et a ainsi innové en mettant sur pied un Centre de compétences dédié non seulement au transfert de compétences tant dans le domaine des TIC, que du management mais aussi aux formations professionnelles.

1. SECTEURS D’ACTIVITES

SECEL demeure l’entreprise dont la vocation est d’être à l’écoute des entreprises pour leur apporter la solution idoine aux problèmes liés aux TIC. La persévérance dont SECEL fait preuve est le signe de la satisfaction que procure chacun des produits et services ; En effet, SECEL est. « **HP** Gold Partner », « **DELL** Business Partner », «**CITRIX** Solution Advisor », **« MICROSOFT** Gold Certified Partner », « **MICROSOFT** Certified Learning Solution », **«** Accredited **EXIN** Center» et Distributeur officiels des produits et solutions **EATON** au Cameroun. A ce titre, SECEL peut mettre à profit la bonne relation qu’elle entretient depuis des années avec ces différents partenaires fournisseurs pour fournir des produits et services dans les conditions les plus avantageuses.

SECEL aujourd’hui, c’est une couverture nationale et internationale, c’est un ensemble de carrières dans plusieurs corps de métier, c’est un point d’honneur mis sur un système de valeur partagées et un engagement pour la satisfaction totale de leurs clients et des autres parties prenantes à travers une qualité croissante de leur prestation de service. Tous ces éléments traduisent une seule ambition :

**Faire de SECEL le premier centre de compétences multidisciplinaires d’envergure internationale en Afrique Centrale**.

Offrir ainsi aux entreprises des solutions innovantes et personnalisées aux problèmes de système d’information et de management, qui répondent aux exigences légitimes de performance, dans la réalisation de leurs ambitions.

Les métiers de SECEL s’articulent autour de trois principaux pôles :

* Protection électrique des onduleurs de grande puissance ;
* ITIL, COBIT et autres bonnes pratiques informatiques ;
* Virtualisation (Citrix, VMware) ;
* Infrastructure d’accès et réseau WAN (Citrix, Packeteer) ;
* Interconnexion et VSAT ;
* Stockage des données et IT Business Continuity ;
* Gestion des projets selon les recommandations du PMI ;
* Les Solutions matérielles et logicielles ;
* La formation et certification ;
* Le conseil et l’assistance **;**
* Son expertise est exprimée à travers 10 domaines de compétences;
* Maintenance du parc informatique ;
* Infogérance matérielle et logicielle ;
* Gestion des projets selon les recommandations

1. LES DEPARTEMENTS

SECEL est une équipe de près de 53 personnes (principalement jeunes) constituée en particulier d’ingénieurs et commerciaux tous diplômés des universités et grandes écoles du Cameroun. Les talents de cette équipe sont organisés en Service. Enumérés comme suit :

1. Un Service Marketing et Commercial

Constitué des hommes et des femmes à votre écoute pour répondre à toutes les sollicitations, mais surtout aider à accroître l’efficacité des actions d’optimisation des systèmes de travail, de renforcement des capacités de vos collaborateurs, à travers leurs conseils.

1. Un service Logistique et Achats

Qui anticipe sur les besoins, et recherche au quotidien la meilleure solution pour répondre à votre objectif de performance au moindre coût.

1. Département Soft and Training

En tant que centre de formation, ce département est spécialisé dans la formation informatique tels que ITIL Foundation, PMP, administration Windows server, Sharepoint, le développement des applications et bien d’autres.

1. Un centre de Service

Doté d’un personnel jeune, dynamique et courtois, qui met à votre disposition son assistance 7j/7et 24H/24 conformément aux normes internationales ITIL.

Ces ingénieurs Techniques et commerciaux sont tous certifiés et spécialisés sur les produits et services proposés aux clients de SECEL. On note ainsi :

* 04 certifiés CITRIX ;
* 07 certifiés HP ;
* 04 certifiés DELL ;
* 02 certifiés McAfee ;
* 01 certifié Packeteer ;
* 03 certifiés CISCO ;
* 02 certifiés mikrotik.
* 06 certifiés MICROSOFT tous MCT (Microsoft Certified Trainer);
* 02 MCPD (Microsoft Certified Professional Developper) ;
* 18 certifiés sur la bonne pratique ITIL ;
* 03 PMP (Project Management Professional) du PMI ;

La jeunesse des équipes confère entre autre souplesse, réactivité, pro activité et disponibilité, autant de qualités requises pour combler les attentes d’une prestigieuse clientèle.

1. LES SOLUTIONS ET LES SERVICES OFFERTS

* **SOLUTIONS MATERIELLES & LOGICIELLES**

**SECEL** offre une diversité de produits, services et formations informatiques et managériales relevant des TIC en partenariat avec de nombreux éditeurs mondiaux pour garantir une qualité absolue. Elle met également à la disposition des clients des systèmes d’exploitation, des outils bureautiques, des outils de développement, des applications spécifiques pour une exploitation optimale de leur matériel.

* **FOURNITURES DES SERVICES INFORMATIQUES**

Elledispose d’une équipe de techniciens qui réalisent l’étude des systèmes informatiques. Cette équipe conseille également dans le choix et l’implémentation des solutions les mieux adaptées à votre entreprise.

* **SECURITE ELECTRONIQUE**

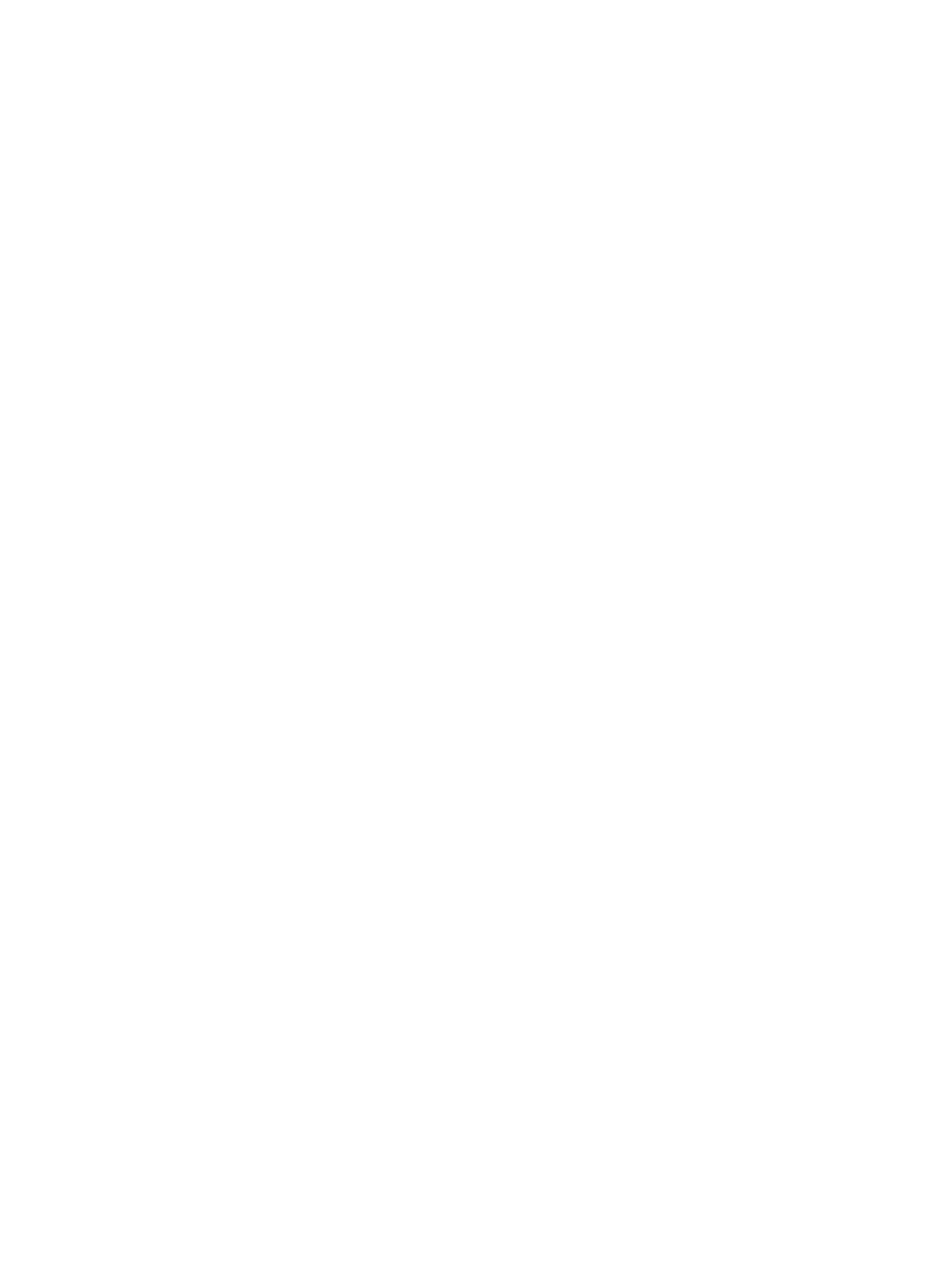
Spécialisée dans la fourniture des solutions de sécurité basées sur les systèmes de vidéo-surveillance, de câblage réseau et d’étude d’installation de satellite VSAT, SECELconçoit, étudie et réalise, tout un ensemble de systèmes électroniques permettant de sécuriser l’accès à des bâtiments et bureaux.

1. ORGANIGRAMME DE SECEL

Figure1 : Organigramme de SECEL Sarl

1. SITUATION GEOGRAPHIQUE

Figure2 : Plan de situation de l’entreprise SECEL



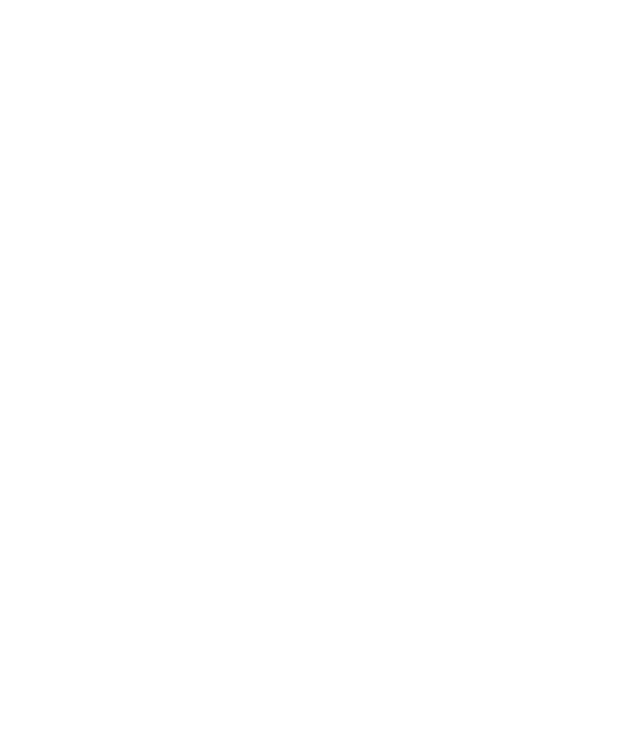
**Douala**



**Yaoundé**



**Garoua**



**SECEL Douala**

**Direction Générale:**

Immeuble I.M.E Situé à 50m de

La mairie de Douala 5

ème

A Bonamoussadi

Tel

:

33 47 88 01

–

Fax

:

99

33 47 87

**Agence de Bali**

606

, boulevard de la République

Carrefour «

Nouvelle route Bonadibong

», Immeuble

R+1, 1

er

étage

BP 15685 Douala

-

Cameroun

Tél

-

fax

(237) 33 43 74 93

:

–

33

64

88

43

E

-

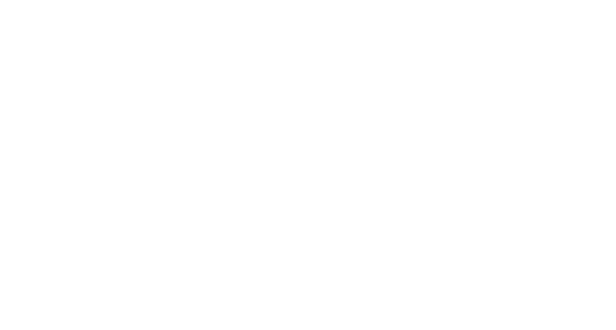
mail

:

INFODLA@SECELGROUP.C

OM

SCC@SECELGROUP.COM



**SECEL Yaoundé**

Omnisport, Derrière TEXACO BP 14828 Yaoundé

Tél

-

fax

:

(237) 22 21 40 08GSM

(

:

237) 99 10

77 56

CT phone

:

(237) 22 01 12 84

–

01 12 82

22

E

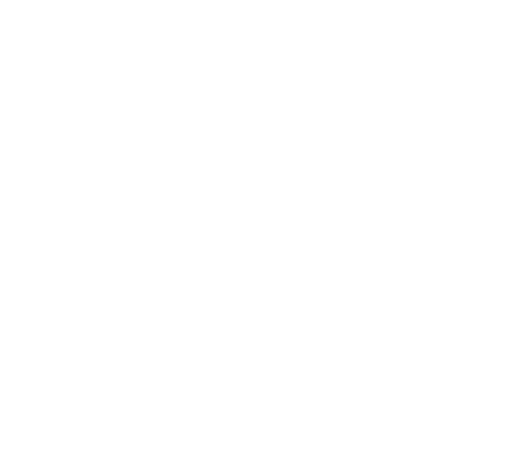
-

mail

:

INFOSYDE@SECELGROUP.

COM



**SECEL Garoua**

**Bidemire**

**Centre Commercial**

Face Direction PMUC

Tél: 22 27 13 83

GSM: 699 68 62 67 6

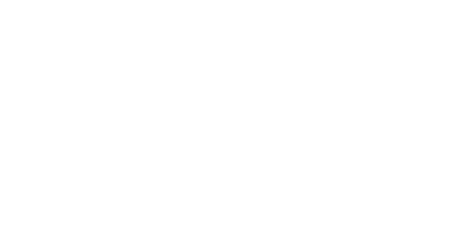
–

19 56 92

77

Email

INFOSGAR@SECELGROUP.COM



**Représentant commercial SECEL**

**Bertoua**

Nkol

-

bikon, Route Nationale N°1

CT Phone

32

22 03 96

:



SECEL Ndjamena



Ndjamena

**Bertoua**

1. FICHE D’IDENTIFICATION ET ORGANISATION

* FICHE D’IDENTIFICATION

Tableau 1 : fiche d’identification de SECEL

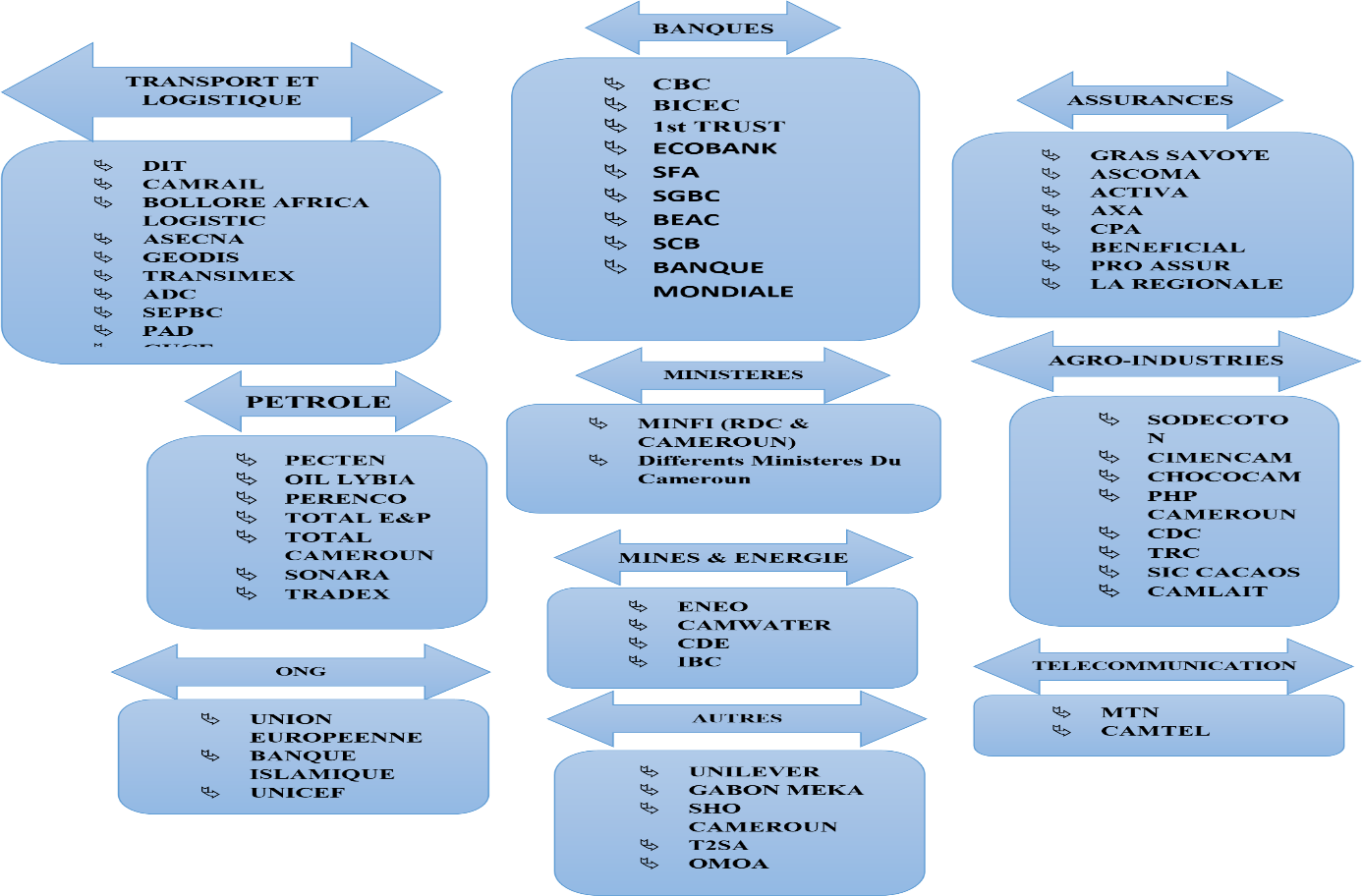
|  |  |
| --- | --- |
| Désignation | Informations |
| Nom de l’entreprise : | SECEL |
| Raison sociale | SARL |
| Capital de la société | 5 000 000FCFA |
| Numéro d’immatriculation | M100100013211L |
| Date et lieu d’immatriculation | 03 OCTOBRE 2001 DOUALA |
| Directeur Général | Blaise OUAFO |
| Secteur d’activité | Informatique & Management |
| Effectifs | Global : 53  Personnel d’encadrement : 08  Personnel technique : 15 |
| Chiffres d’Affaires | 2014 : 1 255 243 307 F CFA  2013 : 740 788 489 FCFA  2012 : 1 225 687 359 FCFA |
| Domaines de compétence | Vente de matériel et logiciels informatiques  Maintenance et sous-traitance informatique  Management informatique  Sécurité informatique et électronique  Certifications internationales  Formations en informatique et management  Développement d’applications |
| Adresse : | **Douala** : 606, Blvd de la République B.P : 15685 Douala-Cameroun ; BP 15 685  **Yaoundé** : Face centre des impôts Biyemassi ; BP 14 828  **Garoua** : Bidemire, Centre commercial face Direction PMUC  **Bertoua** : Nkol-Bikon, Route Nationale N°1  **N’Djamena** : 2ème étage, immeuble ASIA HOTEL, Avenue Behagle Djambalbar |

1. **Vue Externe de SECEL**

On parlera principalement ici des clients, des concurrents et des fournisseurs de SECEL.

1. LES CLIENTS

SECEL dispose d’une multitude de clients qu’on peut regrouper en secteur :



1. LES PARTENAIRES DE SECEL

Suivant les produits et services qu’elle commercialise, SECEL entretient des relations privilégiées avec des partenaires éditeurs, chacun leaders dans leurs domaines dont les principaux sont représentés sur la figure suivante :

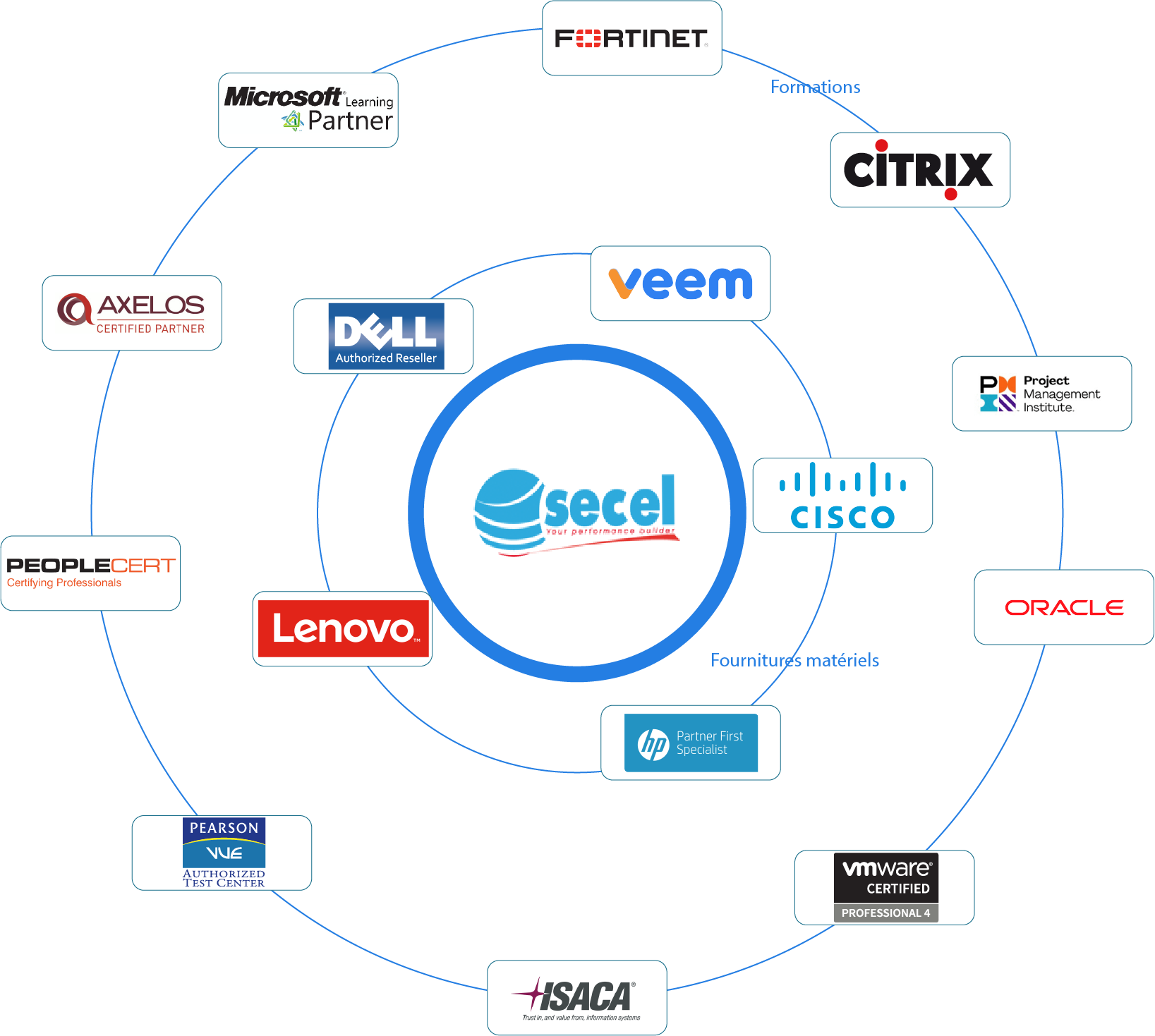


Figure 3 : Liste des Partenaires de SECEL

1. LES CONCURENTS

L’entreprise SECEL a de nombreux concurrents, principalement dans le domaine de la vente du matériel informatique. Parmi ces concurrents les plus influents, on a: WISE COMPUTERS, MIRACLE INFORMATIQUE, JR COMPUTER, COGEICOM, HARDIS, PANESS, France INFORMATIQUE, CFAO.

Maintenant qu’on a parlé de l’entreprise SECEL il sera important pour nous de présenter la manière dont le stage a été effectué.

RESUME

Le présent document expose de manière systématique le déroulement de notre stage au sein de Sofinez ou il nous a été pour travail la mise en place d’un système de réservations hôtelières pour les plateformes mobiles : cas d’ARENA PLAZA.

ARENA PLAZA est une société de service hôtelier qui dispose à son actif de plusieurs agences pour diversifier son activité. Consciente de l’apport des Technologies de l’information et de la Communication dans le domaine professionnel, elle souscrit à un service externe pour vulgariser son activité tant sur le plan national qu’internationale.

Cette expérience motivante et riche en découverte, nous a permis de nous initier aux nouvelles technologies du web et du mobile, et nous a offert l’opportunité de faire un nouveau pas en avant vers l’entrée dans la vie active et dans les métiers IT, que nous retrouverons à la fin de notre formation.

ABSTRACT

This document systematically explains the course of our internship at Sofinez where we worked on the implementation of a hotel reservation system for mobile platforms : case of ARENA PLAZA.

ARENA PLAZA is a hotel service company which has to its credit several agencies to diversify its activity. Aware of the contribution of Information and Communication Technologies in the professional field, it subscribes to an external service to popularize its activity both nationally and internationally.

This motivating experience, rich in discovery, allowed us to be introduced to the new web and mobile technologies, and gave us the opportunity to take a new step towards entering the working life and IT professions, which we will find again at the end of our training.

TABLE DES MATIERES

[DEDICACE I](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839599)

[REMERCIEMENT II](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839600)

[PRESENTATION DE L’ENTREPRISE III](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839601)

[RESUME V](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839602)

[ABSTRACT VII](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839603)

[SOMMAIRE VIII](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839604)

[LISTES DES TABLEAUX XII](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839605)

[LISTES DES FIGURES XIII](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839606)

[LISTES DES ABREVIATIONS XIV](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839607)

[AVANT-PROPOS XV](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839608)

[INTRODUCTION GENERALE 1](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839609)

[PREMIERE PARTIE : ETUDE DE L’ENVIRONNEMENT DE SOFINEZ 2](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839610)

[CHAPITRE 1 : PRESENTATION DE L’ENTREPRISE 3](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839611)

[SECTION 1 : ENVIRONNEMENT INTERNE 4](#_Toc108839612)

[I. HISTORIQUE 4](#_Toc108839613)

[II. MISSIONS DE SOFINEZ 4](#_Toc108839614)

[III. ACTIVITES ET PERSPECTIVES 5](#_Toc108839615)

[A. Activités 5](#_Toc108839616)

[B. Perspectives 5](#_Toc108839617)

[IV. IV. ORGANISATION, FICHE D’IDENTITE ET ORGANIGRAMME 5](#_Toc108839618)

[A. Organisation 5](#_Toc108839619)

[a. Direction générale 5](#_Toc108839620)

[b. Direction des Ressources Humaines 5](#_Toc108839621)

[c. Département Recherche et développement 5](#_Toc108839622)

[d. Département marketing 6](#_Toc108839623)

[B. Fiche d’identité 6](#_Toc108839624)

[C. Organigramme 6](#_Toc108839625)

[SECTION 2 : ENVIRONNEMENT EXTERNE 7](#_Toc108839626)

[V. I. FOURNISSEURS 7](#_Toc108839627)

[VI. II. LES CLIENTS 7](#_Toc108839628)

[VII. III. LA CONCURRENCE 7](#_Toc108839629)

[VIII. 7](#_Toc108839630)

[CHAPITRE2 : ACTIVITE DE STAGE ET PROBLEMATIQUE 8](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839631)

[IX. 9](#_Toc108839632)

[SECTION 1 : ETUDE DE L’EXISTANT 9](#_Toc108839633)

[I. ACTIVITES EN STAGE 9](#_Toc108839634)

[X. II. ETUDE DE L’EXISTANT 9](#_Toc108839635)

[A. Etude d’opportunité 9](#_Toc108839636)

[B. Etude de faisabilité 9](#_Toc108839637)

[C. Description de l’existant 9](#_Toc108839638)

[D. Critique de l’existant 10](#_Toc108839639)

[1. Sur le plan organisationnel 10](#_Toc108839640)

[2. Sur le plan technique 10](#_Toc108839641)

[XI. SECTION 2 : DESCRIPTION DU BESOIN 11](#_Toc108839642)

[XII. 11](#_Toc108839643)

[I. EXTRAIT DU CAHIER DES CHARGES 11](#_Toc108839644)

[A. Présentation générale du projet 11](#_Toc108839645)

[B. Objectifs du projet 11](#_Toc108839646)

[C. Rôle des parties prenantes 12](#_Toc108839647)

[D. Expression des besoins 13](#_Toc108839648)

[1. Les besoins fonctionnels 13](#_Toc108839649)

[2. Besoins non fonctionnelles 14](#_Toc108839650)

[a. Une navigation intuitive et fluide 14](#_Toc108839651)

[b. Le design 14](#_Toc108839652)

[c. La richesse et la qualité de l'information disponible 15](#_Toc108839653)

[d. La fraîcheur de l'information disponible 15](#_Toc108839654)

[e. La réactivité de l’application 15](#_Toc108839655)

[f. L'interactivité de l’application 15](#_Toc108839656)

[g. La visibilité de l’application sur les Stores 16](#_Toc108839657)

[E. Les contraintes du projet 16](#_Toc108839658)

[1. Contraintes d’exploitation 16](#_Toc108839659)

[2. Contrainte de sécurité 16](#_Toc108839660)

[3. Contrainte technique 16](#_Toc108839661)

[F. Déroulement du projet 17](#_Toc108839662)

[a. Plan d'assurance qualité 19](#_Toc108839663)

[b. Documentation 19](#_Toc108839664)

[c. Responsabilités 19](#_Toc108839665)

[DEUXIEME PARTIE : MODELISATION DU SYSTEME FUTUR 20](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839666)

[CHAPITRE 3 : ANALYSE ET CONCEPTION 21](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839667)

[XIII. SECTION 1 : ANALYSE FONCTIONNELLE 22](#_Toc108839668)

[I. Les Acteurs 22](#_Toc108839669)

[II. LES CAS D’UTILISATION 23](#_Toc108839670)

[A. Objectif d’un cas d’utilisation 23](#_Toc108839671)

[B. Identification des Cas d’utilisation 24](#_Toc108839672)

[SECTION 2 : ANALYSE DYNAMIQUE : DIAGRAMME DE SEQUENCE 27](#_Toc108839673)

[XIV. I. LES CONCEPTS DU DIAGRAMME DE SEQUENCE 27](#_Toc108839674)

[XV. II. QUELQUES DIAGRAMMES DE SEQUENCE 28](#_Toc108839675)

[i. Créer un compte 28](#_Toc108839676)

[ii. Se connecter 28](#_Toc108839677)

[iii. Réserver un logement 29](#_Toc108839678)

[SECTION 3 : ANALYSE STATIQUE : DIAGRAMME DE CLASSE 30](#_Toc108839679)

[I. CONCEPTS 30](#_Toc108839680)

[II. DIAGRAMME DE CLASSE DU SYSTEME 32](#_Toc108839681)

[SECTION 4 : CONCEPTION DES DONNEES 33](#_Toc108839682)

[I. MODELE CONCEPTUELLE DES DONNEES 33](#_Toc108839683)

[II. MODELE LOGIQUE DES DONNEES RELATIONNELLES 34](#_Toc108839684)

[III. MODELE PHYSIQUE DES DONNEES 36](#_Toc108839685)

[IV. ARCHITECTURE DE L’APPLICATION 39](#_Toc108839686)

[CHAPITRE 4 : IMPLEMENTATION ET TEST 41](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839687)

[SECTION 1 : IMPLEMENTATION 42](#_Toc108839688)

[I. OUTILS UTILISES 42](#_Toc108839689)

[II. LANGAGES ET TECHNOLOGIES 43](#_Toc108839690)

[III. BASE DE DONNEES 43](#_Toc108839691)

[SECTION 2 : TEST 44](#_Toc108839692)

[CONCLUSION 45](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839693)

[REFERENCES BIBLIOGRAPHIQUES 46](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839694)

[WEBOGRAPHIES 47](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839695)

[TABLE DES MATIERES 48](file:///E:\me\Cours%20Master%202\Projet%20de%20Soutenance\OK\Memoire%20de%20fin%20d'étude.docx#_Toc108839696)

LISTES DES TABLEAUX

[Tableau 1:: Fiche d'identité de Sofinez 5](#_Toc51659437)

[**Tableau 2:Découpage du projet** 15](#_Toc51659438)

[Tableau 3:Tableau descriptif de quelques cas d'utilisation 21](#_Toc51659439)

[Tableau 4:Quelques concepts du diagramme de classes 24](#_Toc51659440)

LISTES DES FIGURES

[Figure 1:Organigramme de Sofinez 5](#_Toc51659412)

[Figure 2: formalisme d'un utilisateur 19](file:///C:\Users\ngami_4vnxadk\Documents\PROJET_STAGE\debut.docx#_Toc51659413)

[Figure 3:formalisme d'un cas d'utilisation 19](file:///C:\Users\ngami_4vnxadk\Documents\PROJET_STAGE\debut.docx#_Toc51659414)

[Figure 4: Diagramme de Cas d'utilisation du système 21](#_Toc51659415)

[Figure 5:Diagramme de séquence :créer un compte 22](#_Toc51659416)

[Figure 6: Diagramme séquence: se connecter 23](#_Toc51659417)

[Figure 7:Diagramme de séquence: Réserver un logement 23](#_Toc51659418)

[Figure 8:Diagramme de Classes du système 25](#_Toc51659419)

[Figure 9: Modèle conceptuelle des Données 26](#_Toc51659420)

[Figure 10:Modèle Physique des Données 29](#_Toc51659421)

[Figure 11: Logos des outils utilisés pour l’implémentation de la solutions 31](#_Toc51659422)

[Figure 12: logos des technologies utilisés pour developper notre solution 32](#_Toc51659423)

LISTES DES ABREVIATIONS

**BLL** : Business Logic Layer

**CIF** : Contrainte d’Intégrité Fonctionnelle

**CIT** : Contrainte d’Intégrité Temporelle

**DAL** : Data Access Layer

**iOS** : iPhone Operating System

**IoT** :

**MCD** : Modèle Conceptuelle des Données

**MLDR**: Modèle Logique des Données Relationnelles

**No SQL** : Base de Données Non Relationnelles

**Sd** : Sequence Diagramm

**UML** : Unified Modeling Language

**UIL** : User Interface Layer

**SDK** : Software Developpnment Kit

INTRODUCTION GENERALE

L'évolution rapide de l'automatisation et le développement rapide de la numérisation sont des statuts auxquels toute entreprise qui se dit sereine doit s'habituer, car il est nécessaire d’entretenir l’information, d’assurer son acheminement et son traitement afin de pouvoir cerner les besoins des clients. De nos jours, de nombreux hôtels proposent des services de réservation en ligne, que ce soit en passant par Booking.com, ou via leur propre site web. Il devient utile de procéder ainsi, limitant les opérations entre le client et la structure hôtelière, de même que les opérations manuelles liées à l’enregistrement des réservations.

C'est dans ce vent de modernité que PREMIUM HOTEL, une société de services hôteliers, souhaite automatiser ses réservations sur son site internet.

**Objectif**

L’objectif de ce projet est de « **mettre en place un système de réservation en ligne intégrée au système d'information de l'entreprise : cas de Premium Hôtel** » afin d’avoir en temps réel au sein d’un même système les informations liées aux réservations. Cette solution aura pour but de permettre à PH d’offrir un service de réservation et de paiement en ligne pour sa clientèle, mais aussi le report en temps réel des données au sein de son système d’information (**HOTIS**).

Le problème qui se pose ici est de savoir **comment faire communiquer Les différents systèmes (site web, système en locale) de Premium Hôtel, de manière sécurisée et d’avoir des données centralisées et en temps réel** ?

**Approche de solution :**

La réalisation de cette solution débute par la compréhension des besoins et attentes utilisateurs en ce qui concerne les réservations. Également la compréhension de l’utilisation du système de réservation existant jusqu’à l’implémentation et le déploiement d’une solution de réservation en ligne. Les principales activités de notre approche sont :

* Le recueil des besoins et attentes des gestionnaires de PH ;
* L’analyse de la proposition de solution à l’aide de diagrammes UML ;
* La conception de l’architecture technique et logicielle, des modèles et base de données et des fonctionnalités de notre solution ;
* L’implémentation de cette solution en 03 composants (un service de réservation via le site web, une application web pour gérer les réservations et la liaison avec le site web, une API pour régir la communication entre l’application web et le système en local pour la visualisation des données en temps réelle).

Afin de gérer notre projet, nous avons utilisé la méthodologie SCRUM, qui est un standard de gestion de projet de développement logiciel. Nous avons également utilisé les meilleures pratiques de l’industrie pour l’ingénierie logicielle et les recommandations de sécurité.

Notre travail sera subdivisé en trois grandes parties :

* **Partie 1 :** dans laquelle nous allons présenter le contexte du projet et la revue de littérature.
* **Partie 2** **:** qui s’articulera autour du matériel et des méthodes utilisées pour l’étude et l’analyse de la solution envisagée.
* **Partie 3 :** il sera question pour nous de présenter d’une part la conception, l’implémentation et test de la solution mise au point et d’autre part faire une conclusion générale sur le travail effectué.

Dans cette partie, nous ferons la description réelle du fonctionnement d’un système d’information, de son rôle pour l’entreprise jusqu’à l’intégration de système d’information. Cette première partie est divisée en deux :

* Le système d’information
* Les généralités sur l’intégration des systèmes d’information.

PREMIERE PARTIE : REVUE DE LITTERATURE

Pour parvenir à offrir à PH une solution de réservation en ligne en temps réel et répondant à ses besoins, il est capital pour nous de bien comprendre les concepts théoriques liés au projet. Ainsi, nous aborderons les notions comme système d’information, intégration de solution, API, paiement en ligne, Bases de données relationnelles….

* 1. **Généralité sur l’intégration de système d’information**
     1. **Système d’information**
        1. **Définition**

Un **système d’information** (SI) est (d’après *Wikipédia*) un ensemble organisé de ressources matérielles, humaines et logicielles permettant de la collecte, le stockage, le traitement, la diffusion ou la distribution de l’information dans un environnement donné.

En outre, il s’agit d’un ensemble matériel ou immatériels (hommes, machines, méthodes, règles) en interaction, transformant en processus des éléments (les entrées) en d’autres éléments (sorties).

* + - 1. **Finalité et rôle d’un Système d’Information (pour l’entreprise)**

Le système d’information tient une importance capitale dans le processus stratégique de toute entreprise. C’est en effet lui qui permet de réaliser toutes les actions liées à la gestion des données : collecte, stockage, traitement et partage… autant d’étapes qu’il rend possible à travers un ensemble de ressources à la fois matérielles, humaines et technologiques. Détenant à la fois un rôle opérationnel et stratégique, le système d’information, au-delà de sa fonction « opérationnelle », dispose d’une véritable dimension sociale en permettant aux différents acteurs d’une entreprise de disposer de l’information dont ils disposent en temps réel, au bon format et diffusable auprès de tous. Lorsqu’il est correctement exploité, le système d’information représente à la fois un gain de temps et d’argent, en rendant automatique des processus jusque-là effectués manuellement, mais il optimise également la performance commerciale en améliorant la relation client.

Le système d’information d’une entreprise est souvent considéré comme sa « colonne vertébrale », à la fois performante et évolutive. C’est pourquoi tout ce qui touche de près ou de loin à sa conception doit faire l’objet d’une réflexion stratégique en amont du lancement du projet.

* + - 1. **Composants d’un SI**

Le domaine des systèmes d'information et de communication a certes une forte composante technologique et [informatique](https://fr.wikipedia.org/wiki/Informatique). Mais c'est seulement un aspect de ce domaine qui est en fait beaucoup plus vaste. Il s'agit de concevoir comment circule et est stockée l'information de façon efficace et cohérente pour toutes les activités d'une entreprise, d'un réseau d'entreprises, d'une administration publique, des relations entre entreprises, des citoyens, des gouvernements, etc.

Ainsi peuvent être inclus comme composants dans un système d’information en vue d’offrir des caractéristiques techniques ou fonctionnalités spécifiques :

* Applications métier ;
* [Bases de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es) de l'entreprise,
* Contrôle d'accès,
* Dispositifs de [sécurité](https://fr.wikipedia.org/wiki/S%C3%A9curit%C3%A9_des_syst%C3%A8mes_d%27information),
* [Infrastructure réseau](https://fr.wikipedia.org/wiki/R%C3%A9seau_(informatique)),
* [Postes de travail informatique](https://fr.wikipedia.org/wiki/Poste_de_travail),
* Accès aux réseaux [Internet](https://fr.wikipedia.org/wiki/Internet), [intranet](https://fr.wikipedia.org/wiki/Intranet) ou [extranet](https://fr.wikipedia.org/wiki/Extranet),
* [Serveurs d'application](https://fr.wikipedia.org/wiki/Serveur_d%27applications),
* Serveurs de [données](https://fr.wikipedia.org/wiki/Donn%C3%A9e) et systèmes de [stockage](https://fr.wikipedia.org/wiki/Sauvegarde),
* Système de paiement électronique,
* Système de sécurité (protection et [chiffrement](https://fr.wikipedia.org/wiki/Chiffrement)),
* Outils de [groupware](https://fr.wikipedia.org/wiki/Groupware), agendas, espace de partage de documents, échange d'informations (forums électroniques), gestion de contacts, conférence électronique (chat, vidéoconférence).
  1. **Intégration de système d’information**

L’**intégration de système** consiste à réunir au sein d’un même système d’information, des parties développées de façon séparées

* + 1. **Présentation de l’intégration de SI**

L'***intégration informatique***, ou ***intégration des systèmes***, désigne le fait de connecter les données, les applications, les API et les appareils au sein d’un service informatique, dans le but d'augmenter l'efficacité, la productivité et l'agilité d’une entreprise. Elle permet à tous les éléments d'un environnement informatique de fonctionner ensemble. Ainsi, elle représente un aspect clé de la transformation d'une entreprise, c'est-à-dire de son adaptation face à l'évolution du marché.

L'intégration ne se limite pas à connecter des éléments, elle ajoute également de la valeur. En effet, en connectant les différentes fonctions de plusieurs systèmes, elle peut offrir de nouvelles fonctionnalités.

***Intégration système différent d’intégration continue***

***L’intégration continue***est unepratique employée par les développeurs, consistant à fusionner des versions de codes qui fonctionnent sur un référentiel central partagé plusieurs fois par jour. L'objectif de l'intégration continue est d'automatiser la création de versions et la vérification afin de détecter rapidement les éventuelles erreurs et d'accélérer le développement.

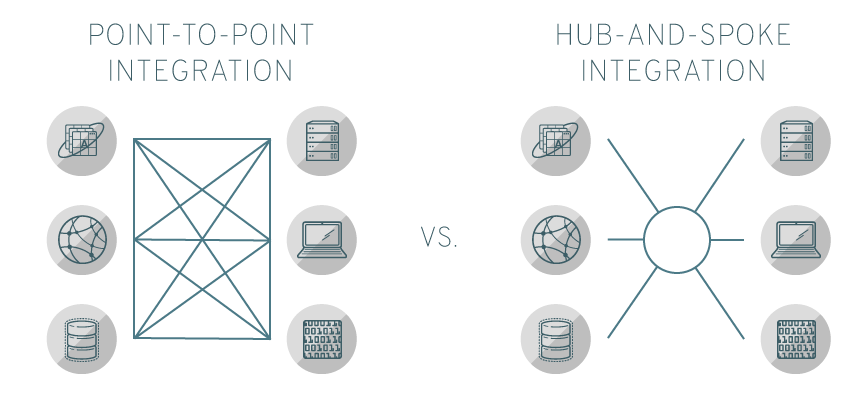
* + 1. **Histoire de l’intégration de SI**

Au fil du temps, les systèmes informatiques grossissaient et se développaient, ils s’éloignaient les uns des autres. Les solutions provenant de fournisseurs différents ne pouvaient plus communiquer entre elles. Au final, les éléments d’une pile informatique n'étaient plus connectés que par le simple fait qu'ils appartenaient à une organisation. Il fallait donc trouver une solution pour organiser l’environnement et éviter la répétition des tâches, surtout lorsqu'il s'agit de mettre en œuvre et de gérer une logique métier.

* + 1. **Les solutions d’intégration de systèmes** 
       1. **L’intégration d’application d’entreprise**

Il existe une solution pour rassembler et organiser tous ces éléments disparates **: l'intégration d'applications d'entreprise** (EAI), c'est-à-dire un ensemble de technologies, des outils et un Framework qui permettent une intégration en temps réel et basée sur les messages entre les différentes applications. L'envoi de ces messages est déclenché par des modifications ou des paramètres au sein de chaque application.

Il existe deux modèles d'EAI : l'intégration point à point (« ***point-to-point*** ») et l'intégration en étoile (« ***hub-and-spoke*** »).

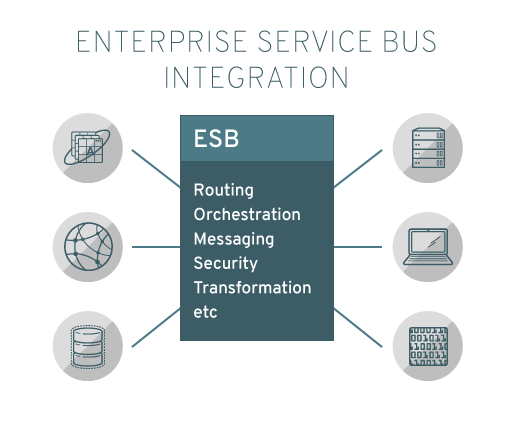


*Point-to-point vs Hub-and-spoke integration*

* **Le modèle point à point :** ici, chaque application (ou ressource) est personnalisée pour pouvoir communiquer (fonctionner) avec les autres applications et éléments de l’environnement informatique. Il s'agit d'une tâche fastidieuse et le modèle est par conséquent fortement sujet aux erreurs. Pour ne rien simplifier, la maintenance du modèle se complexifie à chaque mise à jour de l'infrastructure et des applications.
* **Le modèle étoile :** Il résout le problème du modèle point à point avec un point de connexion central (le cœur) qui relie toutes les applications et les services. Les rayons qui relient le cœur aux applications et aux services peuvent faire l'objet d'une maintenance individuelle. Ainsi, il est possible de concevoir des applications plus spécialisées et réserver les tâches d'intégration au cœur et aux rayons. Le principal désavantage de cette approche réside dans la centralisation du cœur, car il devient le point unique de défaillance du système et des communications au sein de votre infrastructure. Dans un modèle en étoile, toutes les intégrations dépendent, par définition, du bon fonctionnement du cœur.
  + - 1. **ESB**

**ESB** pour *Enterprise Service Bus* est un outil d'abstraction orienté messages qui fournit des modules de service entre les applications.

L'ESB tient également le rôle de point central où tous les modules de service sont partagés, redirigés et organisés afin de connecter les applications et données entre elles. Cette solution remplace avantageusement l'intégration en étoile, mais ce n'est pas non plus la panacée, surtout lorsque l’entreprise croît et acquiert de nouvelles ressources, car elle requiert alors une plus grande rapidité au niveau des propriétés et des ressources logicielles.



*ESB intégration*

De manière structurelle, un ESB ressemble fort au modèle d'intégration en étoile. Mais l'ESB propose tout de même quelques fonctionnalités très spécifiques.

* Un ESB est un service qui utilise des normes ouvertes. Ainsi, pas besoin d'écrire des interfaces uniques pour chaque application.
* Les services d'intégration peuvent être déployés sans modifier fondamentalement les applications.
* Les ESB s'appuient sur des interfaces et des protocoles ouverts et standard, afin de simplifier les nouveaux déploiements.

Toutefois, un déploiement d'ESB implique souvent une architecture centralisée, et ce, pour les mêmes raisons évidentes déjà avancées avec le modèle en étoile : il faut un emplacement unique pour héberger et contrôler tous les services d'intégration. Malheureusement, les déploiements et architectures d'ESB s'accompagnent d'une gouvernance centrale rigide qui ne permet pas de fournir les solutions plus rapides et capables de s'adapter qui sont à la base des initiatives de [transformation numérique](https://www.redhat.com/fr/topics/digital-transformation/what-is-digital-transformation). De plus, les ESB finissent souvent par se changer en applications monolithiques.

* + - 1. **Intégration agile**

Aujourd'hui, les architectures d'applications **cloud-native**, faiblement couplées et développées selon des méthodes [agiles](https://www.redhat.com/fr/devops/what-is-agile-methodology) et des principes DevOps, exigent une approche évolutive et tout aussi agile de l'intégration. L'intégration agile (Selon *Red Hat*) est une stratégie de connexion des ressources qui allie des technologies d'intégration, des techniques de distribution agile et des plateformes natives pour le cloud dans le but d'accélérer la distribution des logiciels tout en renforçant la sécurité. En pratique, l'approche consiste à déployer des technologies d'intégration (des API, par exemple) dans des conteneurs Linux, et à répartir la gestion de l'intégration entre des équipes pluridisciplinaires. Une architecture d'intégration agile peut être divisée en trois capacités clés : **l'intégration distribuée**, les [**conteneurs**](https://www.redhat.com/fr/topics/containers)et les **API**.

|  |  |  |
| --- | --- | --- |
|  | **Caractéristiques** | **Avantages** |
| **Intégration distribuée** | * Encombrement réduit * Basée sur des modèles * Orientée événements * Communautaire | **FLEXIBILITÉ** |
| **Les conteneurs** | * Cloud-native * Simples et déployables individuellement * Évolutifs et hautement disponibles | **ÉVOLUTIVITÉ** |
| **Les API (Application Programming Interface)** | Points de terminaison bien définis, réutilisables et bien gérésInfluence et utilisation de l'écosystème | **RÉUTILISATION** |

En outre ses différents types d’intégration de SI, on distingue des architectures logicielles permettant de faire communiquer plusieurs sous-systèmes ensemble. Notamment :

* L’architecture Orienté évènement (**EDA**) ;
* L’architecture Orienté services (**SOA**) ;
  + - 1. **Architecture orienté évènements**

**L'architecture orientée événements** (de l'anglais *event driven architecture*, ou **EDA**) est une forme d'[architecture de médiation](https://fr.wikipedia.org/wiki/Architecture_de_m%C3%A9diation) qui est un modèle d'interaction applicative mettant en œuvre des services (composants [logiciels](https://fr.wikipedia.org/wiki/Logiciels)) répondant à des sollicitations externes :

* avec une forte cohérence interne (par l'utilisation d'un format d'échange pivot, le plus souvent [XML](https://fr.wikipedia.org/wiki/XML)),
* Et des couplages externes lâches (par l'utilisation d'événements)

Par opposition à l'[architecture orientée services](https://fr.wikipedia.org/wiki/Architecture_orient%C3%A9e_services) (SOA) où un « fournisseur » rend un service à la demande d'un consommateur; en architecture EDA, un [composant](https://fr.wikipedia.org/wiki/Composant_logiciel) prévient par émission d'un événement qu'il a réalisé une opération donnée. C'est aux Clients potentiels de traiter cet événement.

Les architectures EDA ont été popularisées avec l'apparition de standards pour les places de marchés et les systèmes de vente aux enchères. Elles mettent en application une partie des principes d'[urbanisation](https://fr.wikipedia.org/wiki/Urbanisation_(informatique)). Une architecture orientée événements repose principalement sur un bus disposant de fonctionnalités d'abonnement et de publication (*publish and Subscribe*)

***1.2.3.4.1 Les concepts de l’architecture orienté évènements***

Le couplage entre composants est un couplage lâche et les communications sont en général [asynchrones](https://fr.wikipedia.org/wiki/Asynchronisme).

*Le service peut :*

* Être codé dans n'importe quel langage,
* S’exécuter sur n'importe quelle plateforme (matérielle et logicielle).

*Le service doit :*

* S’abonner aux événements qu'il souhaite traiter
* Traiter les événements auquel il est abonné sans préjuger d'un quelconque ordre et émettre un événement compte rendu de l'action qu'il vient de réaliser
* Fournir les événements qu'il est susceptible d'émettre dont les structures sont publiées,
* Être autonome (disposer de toutes les informations nécessaires à son exécution : pas de notion d'état)
* Respecter un ensemble de contrats (règles de fonctionnement).

***1.2.3.4.2 Un évènement***

Un **événement** désigne tout phénomène ou changement d'état significatif au niveau du matériel ou d'un logiciel système. Il ne faut pas confondre un événement et une notification d'événement, c'est-à-dire une notification ou un message envoyé par le système pour signaler à une autre partie du système qu'un événement s'est produit.

Les événements peuvent être causés par des actions internes ou externes. Ils peuvent être provoqués par des utilisateurs (clics de souris ou frappe sur le clavier, par exemple), provenir d'une source externe (un capteur) ou être générés par le système (lors du chargement d'un programme, par exemple).

Les consommateurs d'un événement doivent s'abonner (*subscribe*) à un intermédiaire de gestion d'événement (le bus) et le producteur de l'événement doit le publier auprès de ce gestionnaire (le bus). Quand le gestionnaire d'événement reçoit un événement d'un producteur, il le diffuse (*forward*) aux consommateurs concernés. Si le consommateur est injoignable, le gestionnaire peut conserver le message et le diffuser ultérieurement. Ce moyen de transmission d'événements repose sur un bus de message *store and forward*.

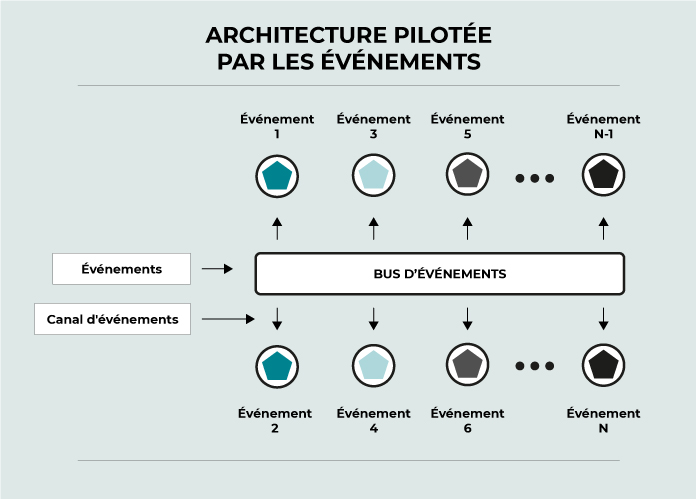
***1.2.3.4.3 Fonctionnement de l’architecture orienté évènements***

Ce type d'architecture implique des **producteurs** et des **consommateurs** d'événements. Un producteur d'événements détecte ou reconnaît un événement et le représente sous forme de message. Il ignore quels seront les consommateurs et les conséquences de chaque événement.

Lorsqu'un événement a été détecté, il est transmis du producteur d'événements au consommateur via des canaux d'événement, où une plateforme de traitement les prend en charge de façon asynchrone. Les consommateurs d'événements doivent être informés lorsqu'un événement se produit. Ils peuvent traiter l'événement ou être seulement affectés par ce dernier.

La plateforme de traitement des événements exécute la réponse adaptée à chaque événement et envoie l'activité en aval aux consommateurs concernés. Cette activité permet de visualiser le résultat d'un événement.

Il existe d'autres gestionnaires d'événements de type middleware qui peuvent servir de plateforme de traitement des événements.



*Architecture Orientée évènements*

***1.2.3.4.4 Modèles d'architectures orientées événements***

Une architecture orientée événements peut être basée sur un modèle de publication/abonnement ou sur un modèle de flux d'événements.

#### **Modèle de publication/abonnement**

Ce modèle est une infrastructure de messagerie basée sur des abonnements à des flux d'événements. Lorsqu'il est utilisé, chaque fois qu'un événement se produit ou est publié, il est envoyé aux abonnés qui doivent en être informés.

#### **Modèle de flux d'événements**

Avec un modèle de flux d'événements, les événements sont enregistrés dans un journal. Au lieu d'être abonnés à un flux d'événements, les consommateurs d'événements peuvent accéder à n'importe quelle partie du flux et le rejoindre à tout moment.

Il existe différents types de flux d'événements :

|  |  |
| --- | --- |
| **Flux d’évènements** | **Description** |
| ***Le traitement des flux d’évènements*** | S’appuie sur une plateforme de diffusion de données en continu, telle qu'Apache Kafka, pour ingérer des événements et des processus, voire transformer le flux d'événements. Le traitement des flux d'événements peut permettre d'y détecter des tendances importantes. |
| ***Le traitement des évènements simples*** | correspond aux cas où un événement déclenche immédiatement une action au niveau du consommateur d'événements. |
| ***Le traitement des évènements complexes*** | oblige un consommateur d'événements à traiter une série d'événements afin de détecter des schémas récurrents. |

**Les flux d’évènements**

**1.2.3.4.5 Avantage de l’architecture orienté évènements**

La programmation orientée événements présente plusieurs avantages, notamment :

* **Des services faiblement connectés**: les créateurs et les consommateurs d’événements ne sont pas connectés. Le créateur ne sait pas quels consommateurs attendent les événements qu’il produit. Il envoie simplement l’événement à un routeur, où les consommateurs pourront trouver les types d’événements auxquels ils sont abonnés.
* **Évolutivité**: comme les créateurs et les consommateurs ne se connaissent pas, il n’y a pas d’intégration directe. Il est beaucoup plus facile d’ajouter de nouveaux services dans le système et d’éviter les pannes au cours de cette opération.
* **Développement agile** : comme il est facile d’ajouter de nouveaux services aux systèmes EDA, le développement s’intègre bien dans un [environnement agile](https://www.lucidchart.com/blog/fr/release-planning-agile). Le routeur d’événements est le médiateur entre les services qui filtre et envoie les événements aux consommateurs.
* **Réduction des coûts** : les systèmes EDA sont basés sur le principe du « push », à la demande. Le système n’interroge pas continuellement les serveurs. La consultation et le traitement à la demande requièrent moins de ressources réseau, ce qui réduit les coûts globaux.
* **Réponse et analyses en temps réel** : les applications réagissent aux changements en temps réel à l’aide des données actuelles et historiques pour prendre des décisions. Ce type de réponse aide le système à prévoir et à éliminer les menaces de sécurité à la volée, en automatisant et en optimisant les chaînes d’approvisionnement.

**1.2.3.4.6 Pourquoi utiliser l’architecture orienté évènements**

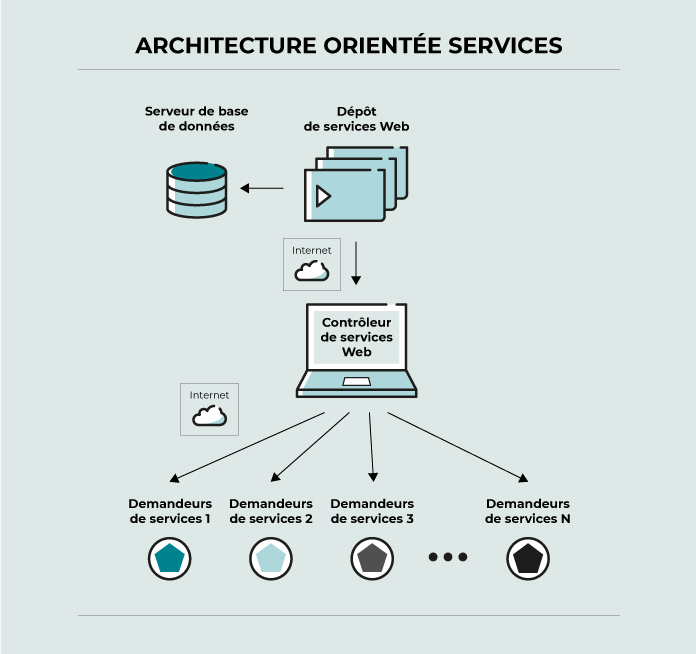
Lorsqu’on souhaite que divers systèmes ou sous-système soient interopérables, l’utilisation d’un EDA est utile pour répondre aux besoins suivants :

* **Réplication des données entre comptes et entre régions :** Les routeurs d’événements peuvent transférer des données vers différents systèmes situés dans différentes régions et différents comptes. Cette approche permet de garantir la synchronisation des systèmes tout en laissant vos équipes développer et déployer de nouvelles applications et de nouveaux services indépendamment des autres équipes.
* **Répartition et traitement parallèle** : utilisez l’architecture EDA pour « répartir » un événement unique entre plusieurs systèmes qui doivent y répondre, ce qui évite à vos équipes de devoir développer un code personnalisé pour transmettre l’événement à chaque consommateur. Chaque système peut traiter l’événement en même temps à des fins différentes.
* **Surveillance et avertissements de l’état des ressources** : les architectures EDA peuvent surveiller les ressources et vous envoyer des alertes en cas de changement d’état, d’anomalies et de mise à jour des composants.
* **Intégration de systèmes hétérogènes** : utilisez des routeurs d’événements pour partager des informations entre des systèmes utilisant différentes piles technologiques.
  + - 1. **Architecture orienté Services (SOA)** 
         1. **Définition**

L'architecture orientée services (ou SOA, *Service-Oriented Architecture*) est un modèle de conception qui rend des composants logiciels réutilisables, grâce à des interfaces de services qui utilisent un langage commun pour communiquer via un réseau.

Un **service** est une unité autonome de fonctionnalité logicielle, ou d'un ensemble de fonctionnalités, conçue pour réaliser une tâche précise comme récupérer des informations ou exécuter une opération. Il contient les intégrations de code et de données nécessaires pour exécuter une fonction métier distincte et complète. Vous pouvez y accéder à distance, et interagir avec lui ou le mettre à jour de manière indépendante.  
En d'autres termes, l'architecture SOA permet à des composants logiciels déployés et gérés séparément de communiquer et de fonctionner ensemble sous la forme d'applications logicielles communes à différents systèmes.

* + - * 1. **Comment fonctionne SOA ?**



*Architecture SOA*

Au sein d'une architecture orientée services, les services communiquent via un système de « couplage faible ». Les composants (aussi appelés « éléments ») sont interconnectés dans un système ou réseau, ce qui leur permet de transmettre des informations ou de coordonner un processus métier, en réduisant la dépendance entre eux.

L’architecture Orienté services est structuré en quatre parties :

* **Le dépôt de services web (Web service repository) :** Il s'agit d'une bibliothèque de services web conçue pour répondre à des demandes d'informations externes. L'information fournie est généralement un petit élément, comme un numéro, un mot, quelques variables, etc. Par exemple, un numéro de vol, un numéro de suivi de colis, le statut d'une commande (une lettre), etc. Cette bibliothèque est généralement documentée de manière très détaillée, car des applications externes font appel aux fonctions qu'elle contient.
* **Le contrôleur de services web (Web service controller) :** Ce module communique les informations contenues dans le dépôt de services web aux demandeurs de services. Lorsqu'un demandeur de service externe appelle une certaine fonction du dépôt de services web, le contrôleur de services web interprète la demande et recherche la fonction dans le dépôt de services web. Il exécute ensuite cette fonction et renvoie une valeur au demandeur.
* **Le serveur de base de données (Database Server) :** Ce serveur contient les tables, les index et les données gérés par l'application. Les recherches et les opérations d'insertion/suppression/mise à jour sont exécutées ici.
* **Les demandeurs de services (Service Requester) :** Il s'agit d'applications externes qui demandent des services au dépôt de services web par l'intermédiaire d'Internet, comme une organisation demandant des informations sur les vols à une compagnie aérienne, ou une autre entreprise demandant à un transporteur la localisation d'un colis à un moment donné.
  + - * 1. **Avantages par rapport à une approche monolithique**
* **Mise sur le marché accélérée et plus grande flexibilité :**le caractère réutilisable des services facilite et accélère le regroupement des applications. Les développeurs ne doivent plus repartir de zéro à chaque fois, comme c'est le cas pour les applications monolithiques.
* **Utilisation de l'infrastructure existante sur les nouveaux marchés :**grâce à l'architecture SOA, les développeurs peuvent plus facilement étendre et mettre à l'échelle les fonctionnalités d'une plateforme ou d'un environnement.
* **Coûts réduits grâce à une meilleure agilité et à un développement plus efficace**
* **Maintenance facilitée :**les services étant autonomes et indépendants, ils peuvent être modifiés et mis à jour autant que nécessaire, sans affecter les autres services.
* **Évolutivité :**comme l'architecture SOA s'adapte à plusieurs services, plateformes et langages de programmation, l'évolutivité est considérablement accrue. En outre, le protocole de communication standardisé limite les interactions entre les clients et les services pour les entreprises, en diminuant le degré d'interaction, les applications peuvent être mises à l'échelle plus facilement et avec moins de pression.
* **Fiabilité améliorée :**il est plus facile de déboguer des petits services plutôt qu'un long code, ce qui permet de créer des applications plus fiables.
* **Grande disponibilité :**les fonctions de l'architecture SOA sont à la portée de tous.
  + - * 1. **Inconvénients de SOA**
* Les services web peuvent représenter une faiblesse au niveau du site pour les pirates informatiques qui veulent obstruer le système. Certaines formes d'attaques sont des « dénis de service ». Elles consistent à demander le même service web des millions de fois par seconde, jusqu'à ce que le serveur tombe en panne. Il existe aujourd'hui une technologie permettant de résoudre ce problème, mais c'est toujours un problème à prendre en compte dans les architectures de services web.
* Le propriétaire du service web aide d'autres sites, mais reçoit une petite rémunération pour ce faire.
  + - * 1. **Quand utilisé SOA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom de l'exemple** | **Suivi de colis** | **Informations sur les vols** | **Convertisseur de devises** |
| **Définition** | Un service web de suivi des colis est une fonction qui, en fonction d’un numéro de suivi, décrit l'itinéraire et la localisation actuelle d'un colis commandé par un client. | Un service web d'information sur les vols accepte en entrée un numéro de vol. Il renvoie les données relatives au vol : heure de départ, heure d'arrivée prévue, lieu de départ, lieu d'arrivée, altitude actuelle, etc. | Un service web de conversion de devises accepte une devise source et donne sa valeur dans une devise de destination (par exemple, « combien de dollars américains font un euro »). |
| **Exemple concret** | * FedEx shipping Integration (fedex.com).   DHL Express Package Tracking (dhl.com). | Flight Explorer Professional Edition (flightexplorer.com) | Toutes les grandes banques en disposent, y compris la banque centrale de nombreux pays. |
| **Avantages** | Les sites de vente au détail qui utilisent ces services web ne doivent pas accéder aux systèmes internes de FedEx ou de DHL.  La fonction de suivi de colis réside sur Internet et peut être utilisée librement par tout site de vente au détail. | Les sites de voyage, les hôtels et les entreprises de logistique qui utilisent ce produit peuvent disposer d'informations immédiates sur un vol par Internet sans avoir à accéder aux systèmes internes. | Les transactions monétaires nécessitent les taux de change officiels à un instant donné. Ce service web de conversion de devises publie toutes les conversions de devises fournies par la Banque centrale. |

* + - * 1. **Architecture SOA ou microservices ?**

Le concept de services introduit par l'architecture orientée services est aujourd'hui un élément essentiel du [cloud computing](https://www.redhat.com/fr/topics/cloud) et de la [virtualisation](https://www.redhat.com/fr/topics/virtualization) dans les solutions de type middleware et microservices.

On confond souvent l'architecture SOA et l'architecture de microservices en raison de leurs similitudes. Toutefois, la principale caractéristique qui les distingue est leur portée : l'architecture SOA est une approche à l'échelle de l'entreprise, tandis que les microservices sont une stratégie de mise en œuvre au sein des équipes de développement des applications.

Ils ne communiquent pas non plus de la même façon avec leurs composants. L'architecture SOA utilise un ESB, alors que les microservices peuvent communiquer entre eux sans état, à l'aide d'[API](https://www.redhat.com/fr/topics/api) indépendantes de tout langage, ce qui permet aux équipes de développement de choisir leurs outils. Ainsi, les microservices offrent plus de tolérance et de flexibilité.

Il arrive également que l'architecture SOA soit confondue avec le modèle [SaaS](https://www.redhat.com/fr/topics/cloud-computing/what-is-saas) (Software-as-a-Service). Le SaaS est une forme de cloud computing qui permet de fournir une application cloud aux utilisateurs, avec ses plateformes et son infrastructure sous-jacentes. Dans une architecture SOA, les services web sont parfois distribués sous forme d'applications SaaS. En général, un fournisseur de services cloud (par exemple AWS, Azure ou IBM Cloud) s'occupe de la gestion de l'environnement cloud sur lequel l'application SaaS est hébergée.

Les utilisateurs interagissent avec le logiciel par le biais d'un navigateur web sur leur ordinateur ou leurs appareils mobiles. Ils peuvent également le connecter à d'autres fonctions à l'aide d'API telles que REST ou SOAP.

* + 1. **Les enjeux liés à l’intégration des systèmes d’information**

L’intégration des systèmes en informatique figure aujourd’hui parmi les principaux enjeux des entreprises. En effet, dans un contexte où les données, les processus et les acteurs sont en perpétuelle évolution, il est parfois bien compliqué de parvenir à « tout faire marcher » ensemble.

Cette difficulté à réunir des éléments très hétérogènes et ne partageant bien souvent que très peu de standards, (en termes de langage de programmation, de format ou encore de [système d’exploitation](https://www.lesjeudis.com/technologies/syst%C3%A8me-d'exploitation) par exemple) met en évidence des problématiques de type technologique, économique et organisationnel auxquelles les entreprises doivent faire face. Le volume de données de plus en plus important ainsi que les [exigences en matière de confidentialité et de sécurité](https://blog.lesjeudis.com/rgpd-la-protection-des-donnees-evolue-en-2018-etes-vous-prets) transforment un peu plus la question de l’intégration des systèmes en un véritable défi.

L’intégration des systèmes permet, à terme, de répondre à différents besoins formulés par les entreprises :

* **Obtenir des processus automatiques** et donc réduire les erreurs pouvant être effectuées lors des processus « manuels » (actions de la part des utilisateurs pour faire converger les différentes applications entre elles)
* **Consolider les données**en étant capable de croiser différents types de données entre elles : en liant les données entre elle, l’entreprise a plus de visibilité en obtenant facilement des statistiques globales
* **Gagner du temps et de l’argent** en regroupant l’ensemble des informations sur un seul et même support (au lieu d’avoir à saisir ces informations sur chacune des applications)

C’est pourquoi, davantage que des enjeux techniques, ce sont bien des [enjeux stratégiques qui se cachent derrière tout projet d’intégration](https://www.supralog.com/technologie-integration-et-enjeux).

Pour l’entreprise, il s’agit de trouver la meilleure équation possible entre les besoins liés à son métier et les capacités logicielles qui sont mises à son service. La réussite d’un tel projet réside en effet en la capacité d’adopter une vision globale tout en faisant intervenir des leviers complémentaires.

Dans cette partie, nous présenterons la démarche que nous avons adoptée dans le cadre de ce travail afin d’arriver à ressortir la solution proposée, ainsi que le matériel utilisé.

Tout d’abord, nus débuterons par la présentation la méthodologie agile de gestion des projets informatiques qui a orienté notre projet. Ensuite, nous irons de la compréhension des besoins et attentes des parties prenantes du projet, à l’analyse des spécifications du projet, à la conception et implémentation de la solution à mettre en œuvre.

DEUXIEME PARTIE : MATERIEL ET METHODES

1. ACTIVITES EN STAGE

Nos activités en entreprise sont basées essentiellement sur :

* La connaissance de l’entreprise et sa vision ;
* Tests fonctionnelles sur PowerH web, sur le site web d’Arena Plaza ;
* L’apprentissage et la manipulation des outils de conception d’image, de logos, de montage d’interface(prototypage) pour nos futurs logiciels (**Web Design**) : à l’instar de Photoshop, Axure RP 9, Filmora9(pour le montage vidéo) ;
* L’apprentissage du langage flutter ;
* La conception des prototypes d’interface mobile pour l’application PowerH pour clients ;

1. II. ETUDE DE L’EXISTANT
2. Etude d’opportunité

Elle consiste à recenser les influences liées au problème et à mettre en place une solution plus efficace. Ainsi le principal problème rencontré est celui de l’indisponibilité de certains clients à se rendre dans une agence hôtelière et d’effectué une réservation, il doit parfois faire recours à un tiers et son besoin n’est pas satisfait.

1. Etude de faisabilité

La mise en place de cette solution informatique aura pour objectif de permettre aux clients de réserver le logement qu’il souhaite, d’avoir des offres promotionnelles et surtout de prévoir, programmer à l’avance un séjour au sein d’une des résidences d’Arena Plaza.

1. Description de l’existant

La plupart des services hôteliers au Cameroun utilisent actuellement un système manuel de réservation de logement, de gestion de la structure hôtelière :

* Le client effectue une réservation téléphonique, mais là souvent les informations du client ne sont pas bien notées, son besoin n’est pas satisfait. Ce qui entraine souvent une mauvaise évaluation des statistiques de la structure.
* Le client se rends dans la structure pour effectuer sa réservation en personne ;
* Les informations relatives à une réservation sont renseignées dans un carnet de réservations ;
* La gestion des comptes client se fait manuellement ;

1. Critique de l’existant

Le système utilisé au sein des structures hôtelières et les processus liés à la gestion des réservations sont pour la plupart manuel.

L’analyse du système actuelle nous a permis de relever les limites suivantes :

1. Sur le plan organisationnel

* La perte de temps ;
* L’accumulation de la paperasse dans le processus de collecte des données
* La lente dans la recherche d’une information précise
* Non prise en compte des spécificités du besoin des clients ;
* La mauvaise gestion des réservations ;

1. Sur le plan technique

* La constitution d’une base de données physique des clients est manuelle, ce qui rends donc sa mise à jour fastidieuse.
* La lenteur dans les traitements des informations clients, pour établir un portefeuille clientèle.
* La quasi-impossibilité d’établir des statistiques fiables sur les réservations effectuées par un client du fait de la proportion importante des données à traiter ;

1. SECTION 2 : DESCRIPTION DU BESOIN
3. EXTRAIT DU CAHIER DES CHARGES

Dans cette partie nous présenterons un extrait du cahier de charge d’ARENA PLAZA.

1. Présentation générale du projet

ARENA PLAZA est une société de service hôtelier qui dispose à son actif de plusieurs agences pour diversifier son activité. Consciente de l’apport des Technologies de l’information et de la Communication dans le domaine professionnel, elle souscrit à un service externe pour vulgariser son activité tant sur le plan national qu’internationale.

Le problème qui réside dans ce mode opératoire est principalement dû à un manque de communication efficace sur son service hôtelier. De plus, le mode opératoire pour la gestion de ce service est très fastidieux (Accumulation de la paperasse, les données ne sont pas centralisées, accès instantané aux statistiques générales sur tous les hôtels de l’organisation).

1. Objectifs du projet

La solution proposée temporairement nommée **PowerH** est une suite de systèmes logiciels inter communicantes à trois composantes :

* Un site web ;
* Une application de gestion en interne ;
* Une application mobile pour consultation des états.
* Une application mobile cliente.

L’objectif de la mise en place d’une application mobile pour clients est de permettre aux clients :

* Effectuer une réservation sans se déplacer ;
* Réserver le logement qu’il souhaite ;
* D’être en contact permanent avec l’hôtel et pouvoir exprimer ses besoins ;
* Avoir un visuel des logements disponibles en temps réel ;

Pour ARENA PLAZ, il s’agit de :

* Augmenter sa présence en ligne ;
* Etablir une relation avec les clients ;
* Améliorer la relation avec le client : le client se sent entouré et écouté
* Améliorer le service client et l’expérience client : le client peut avoir facilement accès aux différents services de l’**hôtel** via l’**application**. C’est plus rapide et plus pratique pour lui ;
* Avoir un avantage concurrentiel : grâce à cette **application** vous pouvez proposer un plus que vos concurrents ne font pas et ainsi donner envie à vos clients de revenir chez vous ;
* Réduire le temps consacré au taches administrative ;
* Augmenter les réservations ;

1. Rôle des parties prenantes

Les parties prenantes du projet sont :

* Equipe de développement :
* Chef de projet : Fokeng Silva ;
* Développeur : Ngami YONWA Michelle MARRION
* Propriétaire : ARENA PLAZA ;
* Utilisateurs : clients ou potentiels clients d’ARENA PLAZA ;

1. Expression des besoins
2. Les besoins fonctionnels

Il sera principalement à caractère vitrine dans la mesure où il communiquera sur tous les services hôteliers fournis par l’organisation.

Sur cette application, nous pourrons dénombrer les fonctionnalités suivantes :

* **Une présentation générale des hôtels de la compagnie** :

Il sera question ici de situer les hôtels, de donner les spécificités de chaque hôtel, présenter les services fournis par ces hôtels (Réservation des chambres, restaurant de l’hôtel, évènements au sein de l’hôtel), présenter l’organisation en générale (personnages clés, partenaires ct…) ;

* **Réservation d’une chambre** :

Cette fonctionnalité permettra à un utilisateur de réserver une chambre pour une durée bien définie et en fonction de la disponibilité des chambres. Cette réservation surgira aussi tôt dans le système interne de gestion hôtelier afin de rendre ladite chambre indisponible dans cette plage horaire.

* **Gestion des comptes clients :**

Cette rubrique est consacrée à l’utilisateur qui se rend sur l’application mobile pour effectuer une réservation. Ses informations de base à l’instar de son nom, numéro de téléphone et son email son pris pour des relances et notifications suite à une réservation. De plus un système de messagerie sera mis en place pour la notification des clients suite à une campagne publicitaire ou un évènement spécial.

* **Gestion des paiements :**

Cette fonctionnalité optionnelle permettra à un client de payer une réservation faite en via son application mobile par **carte visa**, monnaie électronique (pour les clients locaux c’est-à-dire **MOMO**, **OM**) ou **PayPal**. Ce n’est qu’après ce paiement que la chambre de la réservation sera Considérée comme occupée.

1. Besoins non fonctionnelles
2. Une navigation intuitive et fluide

* Le temps d'affichage d'une activité : < 2 s ;
* La rapidité d'accès à l'information : < à 3 clics ;
* L’orientation : savoir à tout moment d'où l'on vient, où l'on va et où l'on peut aller ;
* L’ancrage : revenir facilement à l’accueil ;
* La mise en page : parcourir et exploiter rapidement une activité de l’application sans avoir à chercher à comprendre à quoi elle sert
* La facilité de navigation : doit être homogène sur l'ensemble de l’application

1. Le design

* La qualité du graphisme
* La proportion des interfaces
* L’harmonie des couleurs (celles de l’organisation)
* Le respect d'une charte graphique sur l'ensemble du l’application
* L’adéquation des images avec les valeurs de communication souhaitées

1. La richesse et la qualité de l'information disponible

* La richesse et la qualité des informations techniques de l’application sont des avantages concurrentiels forts.
* Quel que soit la finalité de l’application, les visiteurs attendent une information traitée en profondeur. Un visiteur revient rarement sur une application s'il ne trouve pas l'information recherchée : l'information permet de le fidéliser.

1. La fraîcheur de l'information disponible

* L’application doit être vivante : le contenu de l'information doit être constamment mis à jour et renouvelé afin de pérenniser le trafic sur l’application.
* La présence d'une date de mise à jour est une indication qui permet à l'utilisateur d'être informé et rassuré sur la validité des informations présentées.

1. La réactivité de l’application

* Ce critère permet d'évaluer le temps de réponse de l’application pour une demande d'information, un message envoyé par un utilisateur, ...
* A l'heure de la communication on-line, les clients ne comprendront pas qu'il faille une semaine pour avoir une réponse à un message...

1. L'interactivité de l’application

* "Publier sur le Net, c'est avant tout communiquer".
* La possibilité de réserver ou de commander en ligne, un forum, un dialogue en direct, une newsletter sont des moyens de mettre en place une interactivité entre le site et des visiteurs. Cette interactivité permet notamment à un site de mieux connaître les désirs de ses visiteurs et d'engager une relation de qualité. Elle permet inversement à un visiteur d'être traité individuellement, d'exprimer en ligne ses attentes.

1. La visibilité de l’application sur les Stores

* La visibilité d'une application est souvent associée au référencement de l’application dans les annuaires et moteurs de recherche.
* D'autres moyens contribuent ainsi à la visibilité du site sur le Web : communication offline (journaux, radio, …), publicité on-line (bandeaux, forums), liens vers l’application depuis d'autres sites web. Internet est un réseau mondial d'expression et d'échanges à utiliser.

1. Les contraintes du projet
2. **Contraintes d’exploitation**

L’application sera déployée sur le Play store et l’Apple Store, et mis à la disposition de tous. Pour l’utilisation, l’utilisateur doit disposer d’une connexion internet et d’un téléphone mobile.

1. Contrainte de sécurité

* L’application devra posséder un système d’authentification et d’autorisation ;
* Les données sensibles tel que le numéro de carte visa, le paiement par internet doivent être sécurisés

1. Contrainte technique

L’application sera basée sur une architecture client/serveur, la plateforme de développement utilisé est la suivante :

* **Environnement**: *Android Studio* ;
* **Technologie de développement** : *Flutter/Dart*, patron de conception MVC ;
* **Environnement de stockage** : *FireStore* ;

L’application sera développée sur un ordinateur possédant les caractéristiques suivantes :

* Système d’exploitation : Windows 10 Famille 64bits ;
* Capacité du disque dur : 148 Go ;
* Capacité de la RAM :6,00 Go ;
* Processeur : Intel(R) Core(TM)2 Duo ;

1. **Déroulement du projet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N° | Sprints | Activités et ressources | Date début | Date fin | Qui ? | Livrables |
| 1 | Apprentissage de Flutter | Apprendre flutter/ Dart : <https://flutter.dev/> | 05/08/2020 | 25/08/2020 | Ngami Michelle | Acquisition des connaissances en développement flutter |
| 2 | Benchmarking | Utilisation du cahier de charge :   * Formulation des Objectifs du projet * Recueille de caractéristique du la solution à mettre en place | 26/08/2020 | 26/08/2020 | * M. Fokeng * Ngami Michelle | * Les spécificités fonctionnelles * les caractéristiques ergonomiques. |
| L’analyse du projet :   * Acteurs du système * Cas d’utilisation * Diagramme de cas d’utilisation * Diagramme de séquence système * Diagramme de classe * Diagramme de séquence détaillé | 27/08/2020 | 19/09/2020 | * Ngami Michelle | Cahier d’analyse |
| 3 | Conception Charte graphique / Modélisation | Formation en UX Design : 1.<https://openclassrooms.com/fr/courses/3936801-composez-des-interfaces-utilisateurs-en-material-design>.. 2.<https://openclassrooms.com/fr/courses/3013856-decouvrez-les-fondamentaux-de-l-ux-design>;  3.<https://openclassrooms.com/fr/courses/3013836-initiez-vous-au-design>;  <https://openclassrooms.com/fr/courses/3014016-realisez-le-mock-up-d-une-application-mobile-avec-adobe-xd>.; | 06/09/2020 | 09/09/2020 | * Ngami Michelle | Acquisition des connaissances pour concevoir des interface utilisateurs simples et conviviale respectant les principes de l’UX design |
| Conception des prototypes pour l’application mobile | 09/09/2020 | 13/09/2020 | * Ngami Michelle | Maquettes d’interface de l’application |
| tests prototypes | 14/09/2020 | 15/09/2020 | * M. Fokeng | Adoption d’un nouveau modèle d’interface |
|  |  |  |  |  |

**Tableau 2:Découpage du projet**

1. Plan d'assurance qualité

L’utilisation d’une méthode agile de gestion de projet (scrum) est un facteur qui influence fortement la qualité d’un projet car on se focalise sur les besoins des clients ; On a donc un développement itératif de la solution.

1. Documentation

Comme indique précédent il s’agit de fournir un guide complet d’utilisation et d’installation détaillant l’utilisation d’application.

1. Responsabilités

* Maitre d’ouvrage : Arena Plaza
* Maitre d’œuvre : Sofinez

DEUXIEME PARTIE : MODELISATION DU SYSTEME FUTUR

Dans cette deuxième partie nous allons présenter :

* Analyse et la conception du système futur ;
* Implémentation et test de fonctionnement

CHAPITRE 3 : ANALYSE ET CONCEPTION

L’analyse et la conception du système futur est une étape très importante dans la mise en place d’une nouvelle structure au sein d’une organisation.

Cette analyse et conception se fera grâce au langage de Modélisation **UML**.

1. SECTION 1 : ANALYSE FONCTIONNELLE

Elle désigne une méthode d’organisation du bon sens, consistant à rechercher et à caractériser les fonctions offertes par un système pour satisfaire les besoins d’un utilisateur.

On distingue 2 types d’analyse fonctionnelle :

* L’analyse fonctionnelle externe en vue d’exprimer le besoin de l’utilisateur mettant en évidences les fonctions de services ou d’estime.
* L’analyse fonctionnelle interne qui exprime le point de vue du concepteur du système, mettant en évidences les fonctions techniques.

L’objectif de l’analyse est de définir de manière objective les besoins de l’utilisateur, de recenser toutes les fonctions que doit avoir un produit pour répondre à un besoin et elle s’utilise soit pour concevoir un produit soit pour reconcevoir des produits. Elle s’articule autour de 3 concepts :

* Les Acteurs ;
* Les Cas d’utilisation ;
* Le Diagramme de Cas d’utilisation ;

1. Les Acteurs

Un Acteur est une entité externe qui agit sur le système et jouant un rôle très important dans la délimitation du système du sujet modéliser. Un acteur n’est pas forcément une personne physique et est considéré comme externe aux périmètre d’étude.

21 Il représente un rôle joué par une entité qui interagir avec le système. On distingue deux types d’acteur :

* **Acteur principale** : participant externe interagissant directement avec le système
* **Acteur secondaire** : participant offrant un service qui contribue à la réalisation d’une action ;

Formalisme d’un acteur en UML

« Acteur »

« Nom acteur »

Figure 2: formalisme d'un utilisateur

Dans notre système à modéliser les acteurs sont :

* L’utilisateur (le client) : acteur secondaire ;
* Administrateur (via l’application web) : acteur secondaire ;

1. LES CAS D’UTILISATION

Les Cas d’Utilisation (USE CASE en anglais) est une représentation d’UML du modèle conceptuel, c’est l’image d’une fonctionnalité du système en réponse à une simulation de l’extérieur.

Formalisme d’un cas d’utilisation

Figure 3:formalisme d'un cas d'utilisation

1. Objectif d’un cas d’utilisation

Il a pour objectif de comprendre les besoins du client pour rédiger un cahier des charges fonctionnelle.

1. Identification des Cas d’utilisation

Les différents cas d’utilisation gérer au sein de notre système sont :

* Un utilisateur peut :
  + Se connecter ;
  + Créer un compte ;
  + Modifier compte ;
  + Envoyer un message
* Consulter résidence ;
* Consulter logement ;
* Choisir logement (sélectionner logement)
* Rechercher logement/résidence ;
* Réserver logement ;
* Payer reservation ;
* Modifier les paramètres de l’application (polices, couleurs…) ;
* Administrateur du site web :

Qui s’assure de la mise à jour en temps réel des informations du système (disponibilité des chambres, notifications, système de paiement) ;

**Relations entre cas d’utilisation et Acteur**

|  |  |  |  |
| --- | --- | --- | --- |
| Cas d’utilisation | But | Actions | Acteur |
| Se connecter | Identifier l’utilisateur | Saisir adresse mail et mot de passe | Utilisateur |
| Créer un profil | Accéder au système | Renseigner nom, adresse mail, mot de passe ou utiliser compte Facebook ou twitter |
| Modifier profil | Modifier les information de l’utilisateur | L’utilisateur saisir ces nouveaux informations |
| Envoyer message | Communiquer un besoin à ARENA PLAZA | L’utilisateur saisir un message et l’envoie à Arena Plaza | Utilisateur/système |
| Consulter résidence | Voir les caractéristique d’une résidence | Choisir une résidence | Utilisateur |
| Consulter logement | Voir les caractéristique d’un logement | Choisir le logement à consulter |
| Rechercher logement | Avoir les logement disponible | Saisir le type de logement |
| Sélectionner un logement | Accéder aux caractéristique du logement | Clique sur une image correspondant au logement |
| Réserver logement | Effectuer une réservation | L’utilisateur clique sur réserver | Utilisateur |
| Payer reservation | Payer sa reservation | L’utilisateur choisir le mode de paiement( par carte visa, OM,..) |

Tableau 3:Tableau descriptif de quelques cas d'utilisation

Après description des cas d’utilisation, nous avons ressortir le diagramme de cas d’utilisation suivant :

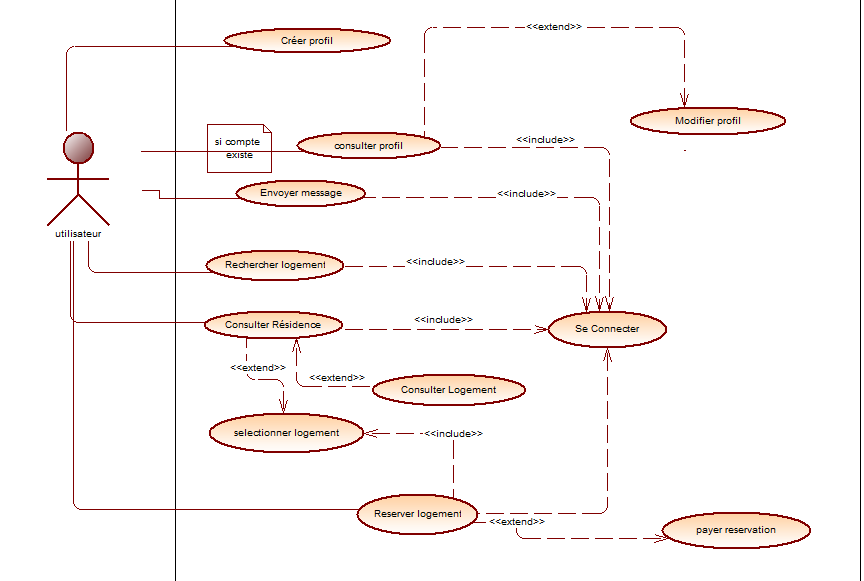


Figure 4: Diagramme de Cas d'utilisation du système

SECTION 2 : ANALYSE DYNAMIQUE : DIAGRAMME DE SEQUENCE

Le diagramme de séquence fait partie des diagrammes comportementaux, plus précisément des diagramme d’interactions. Il permet de :

* Représenter les échanges entre les différents objets et acteurs du système en fonction du temps ;
* D’illustrer, de compléter le Diagramme de Cas d’Utilisation et pour un système donné on peut avoir plusieurs diagrammes de Séquence chacun correspondant à un cas d’utilisation ;

1. I. LES CONCEPTS DU DIAGRAMME DE SEQUENCE

Les principaux concepts du diagramme de Séquence sont :

* Le nom du diagramme précédé de l’étiquette **sd** (sequence diagramm) ;
* L’objet ;
* La ligne de vie : période d’utilisation de l’objet, le moment ou l’objet exécute une méthode ;
* Les messages : communication particulière entre ligne de vie, communication entre objet. Ils peuvent signifier :
* L’innovation d’une opération : appel d’une méthode (message synchrone) :
* L’envoi d’un signal : gestion des évènements, signal (message asynchrone) ;
* Création ou destruction d’une instance de classe ;

1. II. **QUELQUES DIAGRAMMES DE SEQUENCE**
2. **Créer un compte**

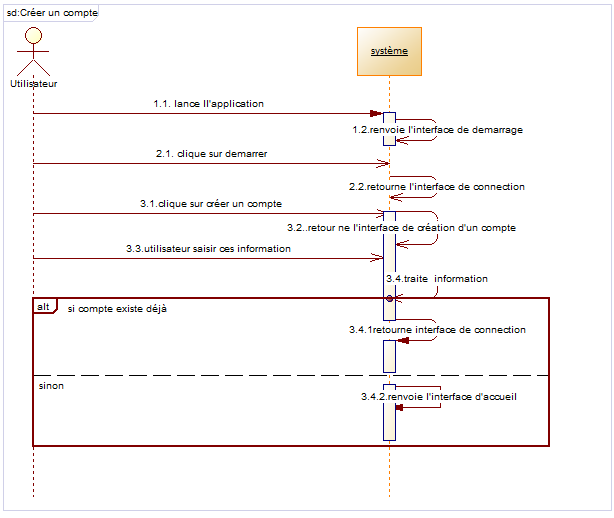


Figure 5:Diagramme de séquence :créer un compte

1. Se connecter

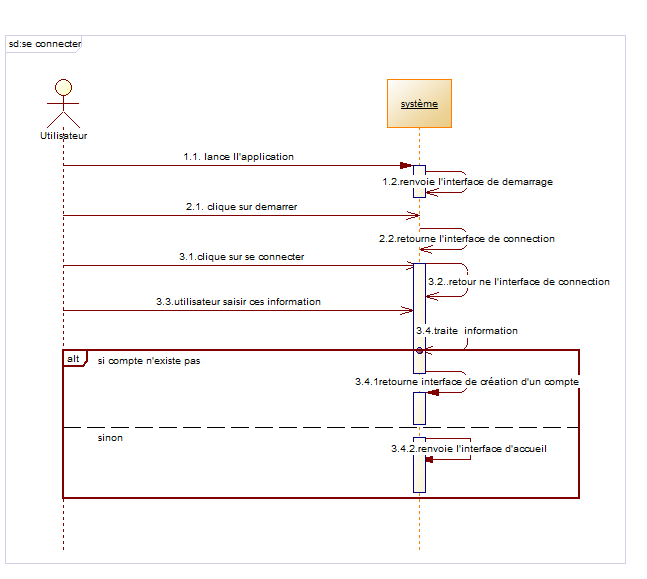


Figure 6: Diagramme séquence: se connecter

1. Réserver un logement

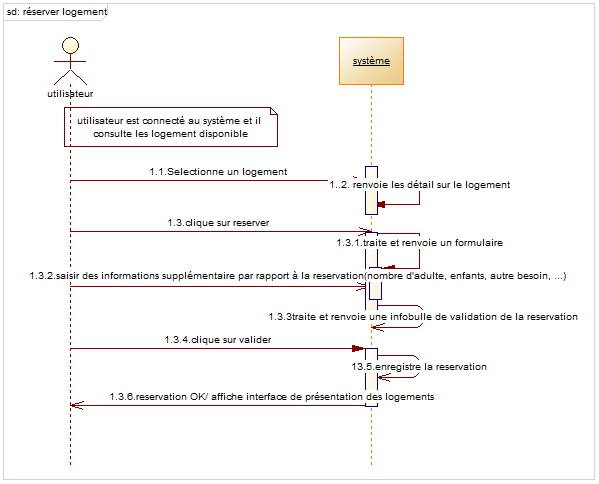


Figure 7:Diagramme de séquence: Réserver un logement

SECTION 3 : ANALYSE STATIQUE : DIAGRAMME DE CLASSE

1. CONCEPTS

Le **diagramme de classes** représente l’architecture conceptuelle du système : il décrit les classes (ou le type des objets) que le système utilise, ainsi que les relations qui les relient entre eux (association, héritage, agrégation, composition,).

Une **classe** est une description d’un groupe d’objets partageant un ensemble commun de propriété (attributs), de comportement (opérations) et de relations avec d’autres objets (associations ou agrégations).

**Formaliste de diagramme de classes**

Les principaux symboles utilisés dans le diagramme de classes sont les suivants :

|  |  |
| --- | --- |
|  | Une note est un commentaire placé sur un diagramme. |
| Classes | Modèle de représentation des objets qui possède des données et des |
| Associations | comportements |
| Attributs | Propriété d'une classe qui décrit un domaine de valeurs possibles partagé par tous les objets de la classe. |
| Opérations | Traitement qui peut être demandé à n'importe quel objet de la classe et partagée par tous les objets de la classe. |
|  | Relations qui pourraient exister entre classe |

Tableau 4:Quelques concepts du diagramme de classes

1. DIAGRAMME DE CLASSE DU SYSTEME

Les différentes classes du système sont :

* Utilisateur(client) ;
* Reservation ;
* Logement
* Catégorie de logement ;
* Résidence ;
* Message ;
* Personnel ;

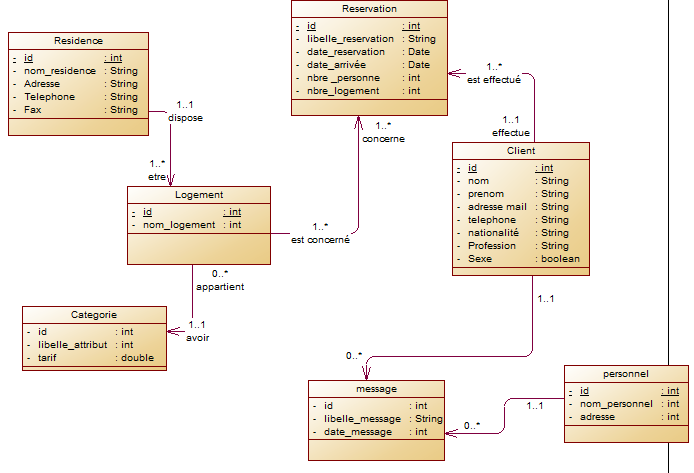


Figure 8:Diagramme de Classes du système

SECTION 4 : CONCEPTION DES DONNEES

1. MODELE CONCEPTUELLE DES DONNEES

Le modèle conceptuel des données représente la structure du système futur BD sans tenir comptes des caractéristiques des supports de stockage.

Le MCD est basé sur le model Entité-association dont les éléments constitutifs sont :

* **Entité** : objet d’une chose abstraite ou concrète pourvu d’existence propre et caractérisée par son unicité dans le SI d’une organisation.
* **Propriété** : rubrique élémentaire permettant de décrire une entité ou une association
* **L’identifiant de l’entité** : propriété particulière de l’entité permettant de distinguer chaque occurrence de l’entité
* **Association** : liens qui unir plusieurs entités.
* **Cardinalité** : couple de valeur qui existe entre chaque entité et son association lié
* **Contrainte d’intégrité fonctionnelle (CIF**) : désignant une dépendance obligatoire entre plusieurs objets participants à une association
* **Contrainte d’intégrité Temporelle (CIT)** : désignant une dépendance temporelle obligatoire entre plusieurs objets participants à une association

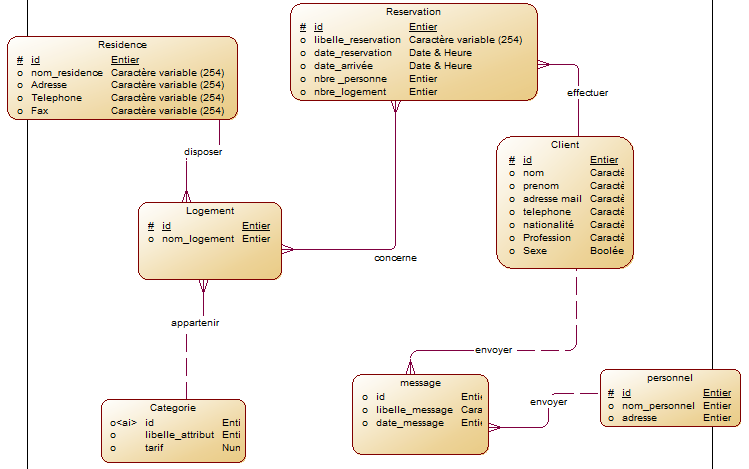


Figure 9: Modèle conceptuelle des Données

1. MODELE LOGIQUE DES DONNEES RELATIONNELLES

Le modèle logique des données relationnel est une représente la base de données sous format directement exploitable par un SGBD. Ces éléments constitutifs du MLDR sont :

* La table
* Clé primaire
* Attribut
* Clé secondaire

Règles de passage du MCD au MLDR :

* L’entité dans le MCD devient Table dans le MLDR, les propriétés attributs, l’identifiant clé primaire de la table.
* Dans une relation père fils la clé primaire du père migre vers la table du fils et devient clé secondaire dans cette table.
* Dans une relation de type plusieurs à plusieurs l’association devient une nouvelle table ayant pour clé primaire la concaténation des clés primaires de l’association.

Suivant ses règles nous avons généré le MLDR suivant :

* UTILISATEUR (id, nom, prénom, adresse\_mail, téléphone, nationalité, profession, sexe) ;
* RESERVATION (id, libelle\_reservation, date\_reservation, date\_arrive, date\_depart, nbre\_personne, #client\_id) ;
* DETAIL\_RESERVATION (id, #id\_logement, #id\_reservation, date, nbre\_logement, nbre\_personne, nbre\_adulte, nbre\_enfant)
* LOGEMENT (id, nom\_logement, #residence\_id, #categorie\_id) ;
* CATEGORIE\_LOGEMENT (id, libelle\_categorie, tarif) ;
* RESIDENCE (id, nom\_residence, adresse, telephone, fax) ;
* MESSAGE (id, libelle\_message, #personnel\_id, #client\_id,) ;
* PERSONNEL (id, nom\_personnel, adresse, telephone...)

1. MODELE PHYSIQUE DES DONNEES

Il s’agit de représenter de façon plus détaillé les caractéristiques des différents champs des tables générées au MLDR :

Structures de quelques tables de la Base de données :

|  |  |  |  |
| --- | --- | --- | --- |
| Relation : *Utilisateur(client)* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| Nom | Chaine de caractère | 254 |  |
| Prenom |
| Adresse\_mail |
| Telephone | Numérique | 20 |  |
| Nationalite | Chaine de caractère | 254 |  |
| Profession |
| Sexe |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation :*Reservation* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| Libelle\_reservation | Chaine de caractère | 254 |  |
| Date\_reservation |
| Date\_arrive |
| Date\_depart | Numérique | 20 |  |
| Client\_id | Int |  | Clé secondaire |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation :*Detail\_reservation* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| id\_logement | Clé secondaire |
| id\_reservation |
| date | date |  |  |
| Nbre\_logement | Numérique | 20 |  |
| Nbre\_adulte | 5 |
| Nbre\_enfant | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation :*Logement* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| nom\_logement | Chaine de caractère | 254 |  |
| Residence\_id | Numérique |  | Clé secondaire |
| Categorie\_id |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation *:Categorie\_logement* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| Libelle\_categorie | Chaine de caractère | 254 |  |
| tarif | Numérique |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation :*Residence* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| nom\_residence | Chaine de caractère | 254 |  |
| Adresse |
| Telephone |
| fax |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation : *Message* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| Client\_id | Clé secondaire |
| Personnel\_id |
| Libelle\_message | Chaine de caractère | 254 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Relation :*Personnel* | | | |
| champs | Type | longueurs | observations |
| Id | Numérique auto |  | Clé primaire |
| Nom\_personnel | Chaine de caractère | 254 |  |
| Adresse |
| Telephone | Numérique | 254 |  |

**Modèle physiques des données**

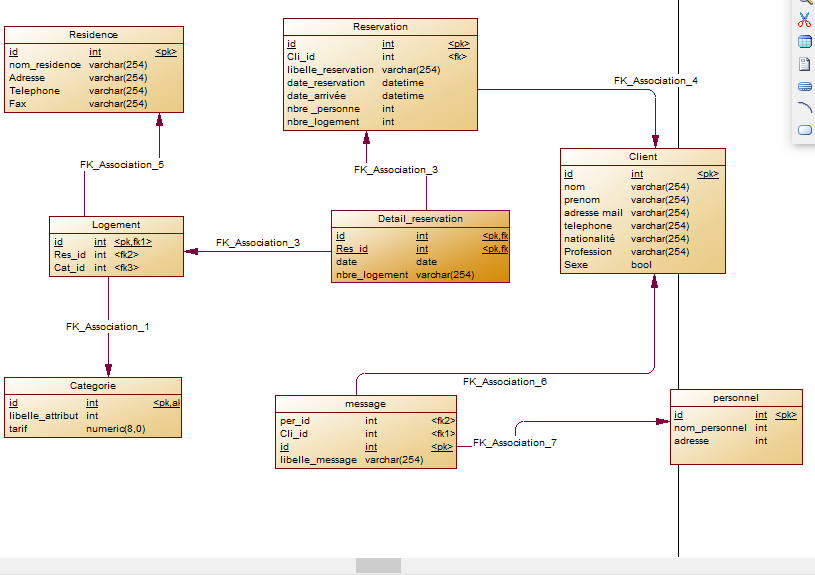


Figure 10:Modèle Physique des Données

1. ARCHITECTURE DE L’APPLICATION

Afin de faciliter la maintenance future et la flexibilité de notre application, nous avons adopté les bonnes pratiques de développement fondées sur l’architecture à trois (03) couches. En effet :

* L’**User Interface Layer** (UIL) correspondqui regroupe l’ensemble des pages visibles au niveau de l’utilisateur ;
* La **Business Logic Layer** (BLL) qui représente la couche métier ou celle qui effectue les traitements.
* La **Data Access Layer** (DAL) est la couche qui se charge d’effectuer les échanges entre l’application et la base de donnéesqui représente les différentes classes métiers de notre application.

Cette séparation permet entre autres :

* De faciliter le développement de l’application avec une meilleure répartition des tâches de conception, et de développement ;
* De bien structurer l’application afin de faciliter sa maintenance et mises à jour ;
* De faciliter les tests de l’application afin de mieux réaliser les tests unitaires, fonctionnels et de non régression ;

CHAPITRE 4 : IMPLEMENTATION ET TEST

Dans cette partie nous allons présenter :

* La phase d’implémentation : où nous parlerons des outils utilisés pour la réalisation du système à mettre en place ;
* La phase de test : où nous présenterons quelques interfaces de notre application

SECTION 1 : IMPLEMENTATION

1. OUTILS UTILISES

Pour mener à bien ce projet, nous nous proposons d’utiliser les méthodes et outils de développement hautement appropriés à la problématique suscitée :

* **UML**, Comme langage d’analyse et de modélisation qui nous a permis de bien spécifier les besoins, de les analyser et enfin de concevoir la solution retenue ;
* **Android Studio** : [environnement de développement](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement) pour développer des applications mobiles [Android](https://fr.wikipedia.org/wiki/Android). Il est basé sur [IntelliJ IDEA](https://fr.wikipedia.org/wiki/IntelliJ_IDEA) et utilise le [moteur de production](https://fr.wikipedia.org/wiki/Moteur_de_production) [Gradle](https://fr.wikipedia.org/wiki/Gradle). Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément
* **PowerAMC :** logiciel de modélisation
* **Axure RP 9 :** pour la conception des interfaces ;
* **FireBase de FireStore :** systèmede stockage de données en ligne, le langage de gestion des données utilisés est le No SQL











Figure : Logos des outils utilisés pour l’implémentation de la solutions

1. LANGAGES ET TECHNOLOGIES

Les Langagues de développement utilisés sont :

* **Flutter** :  [kit de développement de logiciel](https://fr.wikipedia.org/wiki/Kit_de_d%C3%A9veloppement) (SDK) d'interface utilisateur OpenSource créé par Google. Il est utilisé pour développer des applications pour [Android](https://fr.wikipedia.org/wiki/Android), [iOS](https://fr.wikipedia.org/wiki/IOS), [Linux](https://fr.wikipedia.org/wiki/Linux), [Mac](https://fr.wikipedia.org/wiki/MacOS), [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows), [Google Fuchsia](https://fr.wikipedia.org/wiki/Google_Fuchsia)  et le [web](https://fr.wikipedia.org/wiki/World_Wide_Web) à partir d'une seule base de code.
* **Dart :** Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web. Dart est un langage orienté objet, basé sur la classe, récupérateur de mémoire avec une syntaxe de type C.





Figure : logos des technologies utilisés pour developper notre solution

1. BASE DE DONNEES

Comme système de stockage nous avons utilisés **Google cloud : FireStore** de FireBase

* **Firebase** est un ensemble de services d'hébergement pour n'importe quel type d'application ([Android](https://fr.wikipedia.org/wiki/Android), [iOS](https://fr.wikipedia.org/wiki/IOS_(Apple)), [JavaScript](https://fr.wikipedia.org/wiki/Javascript), [Node.js](https://fr.wikipedia.org/wiki/Node.js), [Java](https://fr.wikipedia.org/wiki/Java_(technique)), [Unity](https://fr.wikipedia.org/wiki/Unity_(moteur_de_jeu)), [PHP](https://fr.wikipedia.org/wiki/PHP), [C++](https://fr.wikipedia.org/wiki/C%2B%2B) ...). Il propose d'héberger en [NoSQL](https://fr.wikipedia.org/wiki/NoSQL) et en temps réel des bases de données, du contenu, de l'[authentification sociale](https://fr.wikipedia.org/wiki/Social_login) ([Google](https://fr.wikipedia.org/wiki/Google), [Facebook](https://fr.wikipedia.org/wiki/Facebook), [Twitter](https://fr.wikipedia.org/wiki/Twitter) et [GitHub](https://fr.wikipedia.org/wiki/Github)), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.
* Cloud **Firestore** est une base de données entièrement gérée et native au cloud qui facilite le stockage, la synchronisation et l'interrogation des données pour les applications Web, mobiles et IoT. Il utilise le NoSQL pour gérer les données

SECTION 2 : TEST

CONCLUSION

Le développement d’application, de logiciels demande beaucoup d’attention et de méthodes, permettant ainsi de mettre au point une solution pouvant palier a un problème bien déterminé. L’automatisation, est un atout indispensable en cette ère de la mondialisation ou il ne sera plus question de tout gérer manuellement mais de s’aider d’équipement informatique et logiciels. Le travail qui nous a été demandé pendant notre stage chez Sofinez était la mise en place d’un système de réservations hôtelières pour plateformes mobiles : cas d’ARENA PLAZA. Cette nouvelle solution devrait aider les clients d’ARENA PLAZA à effectuer une réservation sans avoir à se déplacer, et de disposer du logement qu’il souhaite.

L’étude de ce projet nous a permis de renforcer nos connaissances dans le domaine du développement mobile et d’améliorer nos facultés de recherche et de détermination.

Nous n’avons pas la prétention d’avoir réalisé une œuvre parfaite c’est pourquoi nous vous remercions d’avance tous ceux qui nous ferons parvenir des critiques et suggestions constructives.

REFERENCES BIBLIOGRAPHIQUES

* Cours « UML » de M. Mbanjock, enseignant à IME 2018-2019 (Non publié) ;
* Cours « UML », « Ingénierie logiciel » de M. Kenfack, enseignant à IME 2019-2020 (Non publié) ;
* Cours « Développement mobile » de M. Fokeng, enseignant à IME 2019-2020 (Non publié) ;
* **Rapports de stage** :
* *« Mise en place d’une application de test de niveau relative aux examens Internationaux de la langue anglaises : cas de VLC »*, rédigé et soutenu par MINOU DJEUKI PASCAL, 2018-2019 ;
* « *Développement du module des Consultations hospitalières : cas de WES@IT »*, rédigé et soutenu par TALLA TAGNE PASCAL BLAISE, 2018-2019 ;
* *« Automatisation de la gestion des ressources : cas de IME* », rédigé et soutenu par FOKENG SILVA, 2017-2018 ;

WEBOGRAPHIES

* <https://openclassrooms.com/fr/courses/3013856-ux-design-decouvrez-les-fondamentaux>;
* <https://openclassrooms.com/fr/courses/3936801-composez-des-interfaces-utilisateurs-en-material-design>;
* <https://openclassrooms.com/fr/courses/3938146-apprehendez-la-demarche-ux-en-pratique>;
* <https://pub.dev/packages/cloud_firestore>;
* <https://flutter.dev/docs/development/ui>;
* <https://api.flutter.dev/flutter/material/material-library.html>;
* <https://github.com/flutter/website/blob/master/examples/layout/lakes/step2/lib/main.dart>;
* <https://www.pinterest.com>;
* <https://www.pinterest.fr>;
* <https://material.io/resources/tutorials>;
* <https://material.io/resources/color>;
* <https://fonts.google.com>;
* <https://docs.axure.com/axure-rp/tutorials>;

TABLE DES MATIERES

DEDICACE I

REMERCIEMENT II

SOMMAIRE III

LISTES DES TABLEAUX IV

LISTES DES ABREVIATIONS V

AVANT-PROPOS VI

RESUME IX

ABSTRACT X

INTRODUCTION GENERALE 1

PREMIERE PARTIE : 2

ETUDE DE L’ENVIRONNEMENT DE SOFINEZ 2

CHAPITRE 1 : PRESENTATION DE L’ENTREPRISE 3

SECTION 1 : ENVIRONNEMENT INTERNE 3

I. Historique 3

II. Les missions de SOFINEZ 3

III. Les activités et perspectives 4

A. Les activités 4

B. Les perspectives 4

IV. Organisation, fiche d’identité et organigramme 4

A. Organisation 4

a. Direction générale 4

b. Direction des Ressources Humaines 4

c. Département Recherche et développement 4

d. Département marketing 4

B. Fiche d’identité 5

C. Organigramme 5

SECTION 2 : ENVIRONNEMENT EXTERNE 5

I. Fournisseurs 5

II. Les clients 6

III. La concurrence 6

CHAPITRE 2 : ACTIVITE DE STAGE ET PROBLEMATIQUE 7

SECTION 1 : ETUDE DE L’EXISTANT 7

I. Activités en stage 7

II. Etude de l’existant 7

A. Etude d’opportunité 7

B. Etude de faisabilité 7

C. Description de l’existant 8

D. Critique de l’existant 8

1. Sur le plan organisationnel 8

2. Sur le plan technique 8

SECTION 2 : DESCRIPTION DU BESOIN 8

I. Extrait du cahier des charges 8

A. Présentation générale du projet 9

B. Objectifs du projet 9

C. Rôle des parties prenantes 9

D. Expression des besoins 9

1. Les besoins fonctionnels 10

2. Besoins non fonctionnelles 10

a. Une navigation intuitive et fluide 10

b. Le design 11

c. La richesse et la qualité de l'information disponible 11

d. La fraîcheur de l'information disponible 11

e. La réactivité de l’application 11

f. L'interactivité de l’application 11

g. La visibilité de l’application sur les Stores 12

E. Les contraintes du projet 12

1. Contraintes d’exploitation 12

2. Contrainte de sécurité 12

3. Contrainte technique 12

DEUXIEME PARTIE : 14

MODELISATION DU SYSTEME FUTUR 14

CHAPITRE 3 : ANALYSE ET CONCEPTION 15

CHAPITRE 4 : IMPLEMENTATION ET TEST 16

SECTION 1 : IMPLEMENTATION 16

SECTION 2 : TEST 16

CONCLUSION 17

REFERENCE BIBLIOGRAPHIQUE 18

WEBOGRAPHIE 19

TABLE DES MATIERES 20