REPUBLIQUE DE CÔTE D'IVOIRE

Union-Discipline-Travail

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Universite Nangui Abrogoua

UNITE DE FORMATION ET DE RECHERCHE DES SCIENCES FONDAMENTALES ET APPLIQUEES

2016 - 2017





Thème:

CONCEPTION ET REALISATION D'UN SYSTEME DE GESTION DES ACTIVITES DES COMMERCIAUX DANS UNE STARTUP :

Cas de l'entreprise XIKKA

Mémoire pour l'obtention du grade de MASTER

Option Méthodes Informatiques Appliquées à la Gestion et à l'Economie (MIAGE)

Présenté et soutenue par :

KOFFI KASSI ÉRIC

Le jury Directeur de mémoire :

M. AKA Boko, Professeur titulaire, UNA

Président du jury : Co-directeur :

Membre:

M. N'TAPKE Tchimou, Maitre-Assistant, UNA

Membre:

DEDICACES

Je dédie ce travail

A mon tuteur KOUAME EMMANUEL.

A mon père KOUA EMMANUEL.

A ma mère KOUAME AUGUSTINE.

A mes chères frères et sœurs.

A tous ceux qui ont cru en moi et qui me donnent l'envie d'avancer.

REMERCIEMENTS

Durant mon stage, j'ai bénéficié du soutien moral et technique de plusieurs personnes ce qui a rendu mon environnement de travail très agréable.

Je remercie tout d'abord Monsieur et Madame AKEYA pour m'avoir permis de réaliser ce stage et pour m'avoir guidé et encadré au cours de celui-ci dans de très bonnes conditions et dans une ambiance de travail très agréable.

J'adresse également tous mes remerciements à :

- Tout le corps enseignant de l'UFR SFA, particulièrement mon directeur de mémoire le Docteur N'TAKPE TCHIMOU qui a pu favoriser ce travail,
- Mon responsable de filière, Docteur EDI HILAIRE pour tous ces efforts fournis à l'égard des étudiants de la filière MIAGE,
- Tout le personnel de l'entreprise Xikka pour son accueil chaleureux, son soutien tout au long de mon stage et les diverses connaissances qu'ils ont partagées avec moi durant toute cette période,
- Mes camarades de Master MIAGE pour leurs conseils avisés et en particulier KALO WILLIAMS ET YAO AUBIN,

SOMMAIRE

DED	ICACES	i
REM	IERCIEMENTS	ii
LIST	E DES FIGURES	iv
LIST	E DES SYMBOLES	v
RES	UME	vii
ABS [*]	TRACT	viii
	RODUCTION	
	PITRE I : PRESENTATION GENERAL DE L'ENVIRONNEMENT	
1.	L'entreprise	3
СНА	PITRE II : CONTEXTE GENERAL DU PROJET	6
1.	•	
2.	Présentation du projet	7
3.	Objectifs du projet	8
СНА	PITRE III: CONCEPTION ET REALISATION DE L'APPLICATION	9
I.	CONCEPTION	10
1.	Diagramme des cas d'utilisations	10
2.	Diagramme d'activité	11
3.	Diagramme des classes	15
II. I	PHASE DE REALISATION	16
1.	Choix des technologies	16
2.	Architecture et fonctionnement de l'application	28
3.	Conclusion partielle	30
СНА	PITRE IV : PRESENTATION DE L'APPLICATION OBTENUE	31
1.	Les interfaces	32
2.	La sécurité de l'application	43
CON	ICLUSION	44
RÉFI	ÉRENCES BIBLIOGRAPHIQUES	45
ТЛР	DI E DEC MATIEDEC	45

LISTE DES FIGURES

Figure 1-Logo Xikka	3
Figure 2-Organigramme Xikka	5
Figure 3- Diagramme des cas d'utilisation	10
Figure 4- Diagramme d'activité « Authentification »	12
Figure 5- Diagramme d'activité « Créer Profil Utilisateur »	13
Figure 6- Diagramme d'activité « enregistrer des Clients »	14
Figure 7- Diagramme des classes	15
Figure 8- Langages de programmation	17
Figure 9- Frameworks PHP	20
Figure 10- Classement de popularité des SGBD	25
Figure 11- Structure des fichiers sous Laravel	28
Figure 12- Interface authentification	32
Figure 13- Page accueil commercial	33
Figure 14- Page accueil administrateur	33
Figure 15- Interface d'enregistrement client	34
Figure 16- Interface nombre d'enregistrement plusieurs clients	35
Figure 17- Interface d'enregistrement plusieurs clients	36
Figure 18- Liste des clients	36
Figure 19-Interface d'envoi Email	37
Figure 20-Interface RDV	38
Figure 21-Interface reporté RDV	39
Figure 22-Liste des RDV	39
Figure 23- Détails RDV reporté	40
Figure 24-Interface de création profil utilisateur	40
Figure 25-Liste des clients d'un commercial choisi	41
Figure 26-Détails client	42
Figure 27- Interface bloquer/débloquer utilisateur	42

LISTE DES SYMBOLES

CSS: Cascading Style Sheets. Feuilles de style en cascade, forment un langage informatique qui décrit la présentation des documents HTML et XML.

HTML: 'HyperText Markup Language, est le format de données conçu pour représenter les pages web. C'est un language de balisage permettant d'écrire de l'hypertexte.

HTTP: Hypertext Transfer Protocol en français protocole de transfert hypertexte.

LDAP: Lightweight Directory Access Protocol est à l'origine un protocole permettant l'interrogation et la modification des services d'annuaire.

MIAGE: Méthodes Informatiques Appliquées à la Gestion et à l'Economie.

MVC: Model Vue Contrôleur. Une architecture de programmation de logiciel.

ONG: Organisation Non Gouvernementale.

ORM: Object Relation Mapping est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle.

PHP: Hypertext Preprocessor, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur http.

POP: Post Office Protocol.

SFA: Sciences Fondamentales et Appliquées.

SGBD : Système de Gestion de Base de Données.

SQL: Structured Query Language, en français langage de requête structurée est un langage informatique normalisé servant à exploiter des bases de données relationnelles.

SSH: Secure Shell. C'est un programme informatique et un protocole de communication sécurisé.

UFR: Unité de Formation et de Recherche.

UML : Unified Modeling Language en français Langage de Modélisation Unifié.

XML: L'Extensible Markup Language (langage de balisage extensible en français).

RESUME

Ce travail consiste à la conception et à la réalisation d'une application web pour la Gestion des activités des commerciaux de l'entreprise Xikka. Cette application permettra à Xikka d'enregistrer, de traiter les informations relatives aux clients et aux prospects, dans le but de les fidéliser en leur offrant ou proposant des services.

Pour la réalisation de ce projet, nous avons utilisé UML (langage de modélisation unifié) comme langage de modélisation. PHP (Hypertext Preprocessor) est le langage de programmation choisi avec son Framework Laravel et POSTGRES comme système de gestion de base de données (SGBD).

Mots clés: Xikka, UML, PHP, POSTGRES, SGBD

ABSTRACT

This work consists in the conception and to realize It of a Web application for

the management of the commercial activities of the company Xikka. This application

will allow Xikka to register, to handle the data relative to the customers and to the

prospects, in order to retain them by offering them or providing services.

For the realization of this project, we used UML (Unified Language of Modeling) as

language of Modeling. PHP (Hypertext Preprocessor) is the programming language

chosen with its Framework Laravel and POSTGRES as the database management

system (DBMS).

Keywords: Xikka, UML, PHP, POSTGRES, DBM

INTRODUCTION

Le monde de l'entreprise est économiquement difficile. Dans ce contexte, la relation de l'entreprise avec ses clients est un enjeu stratégique majeur. Nos clients, c'est notre chiffre d'affaires! Affirment les chefs d'entreprises. C'est pourquoi ils sont la clé de la pérennité et du développement d'une entreprise.

Les technologies de l'information et de la communication sont des moyens pour les entreprises de mieux connaître leur clientèle et de gagner leur fidélité en utilisant les informations dont elles disposent, de manière à mieux cerner leurs besoins et donc de mieux y répondre.

Fidéliser un client coûte beaucoup moins cher que trouver de nouveaux clients par la prospection. C'est la raison pour laquelle un grand nombre d'entreprises orientent leur stratégie autour des services proposés à leurs clients.

Les sources d'informations sur les clients sont souvent dispersées dans l'entreprise et gérées dans différents logiciels. Il est donc intéressant de se doter d'un outil permettant de collecter et de structurer l'information pour améliorer la productivité des équipes commerciales, marketing et autres services impliqués dans le cycle de vente.

Dans ce rapport, nous allons vous présentez les différentes étapes qui ont permis la conception et la réalisation d'une application de Gestion des activités des commerciaux.

CHAPITRE I:

PRESENTATION GENERAL DE L'ENVIRONNEMENT

1. L'entreprise



Figure 1- Logo Xikka

XIKKA est une SA (Société Anonyme) spécialisée dans le domaine des transactions financières au capital de 25 000 000 FCFA. Son siège est situé à Angré 8ème Tranche, Abidjan.

1.1. Services

Les différents services qu'offre Xikka sont :

✓ Approvisionnement de portemonnaie web

Xikka dispose d'une plateforme (<u>www.xikka.net</u>) dont toute personne peut se connecter sur cette plateforme et créer son compte (portemonnaie web). Ces personnes peuvent approvisionner leur compte Xikka en se rendant dans une agence Xikka.

✓ Envoi et réception d'argent

Les clients de Xikka ont la possibilité de transférer de l'argent à quelqu'un à partir de leur compte Xikka ou de recevoir de l'argent dans leur compte Xikka.

✓ Approvisionnement de comptes bancaires

Les clients de Xikka peuvent approvisionner leur compte bancaire à partir de leur portemonnaie web Xikka.

√ Paiement de factures

La possibilité de payer les factures CIE, SODECI à partir de leur compte Xikka.

✓ Achat en ligne

La possibilité de faire leur achat en ligne à partir de leur portemonnaie web Xikka.

✓ Recharge de crédit

Les clients ont la possibilité de se recharger en crédit de communication tous réseaux téléphoniques présents sur le territoire ivoirien.

✓ Dépôt sur compte mobile money

Les clients ont la possibilité d'approvisionner leur différent compte mobile money tous réseaux téléphoniques présents sur le territoire ivoirien

✓ Collecte de fonds

La possibilité de collecter de l'agent dans leur compte Xikka.

✓ Paiement de masse

Les entreprises clientes de Xikka ont la possibilité de payer leurs employés via la plateforme Xikka.

✓ Rechargement de cartes bancaires prépayées

La possibilité de recharger des cartes bancaires et prépayées (Carte UBA) à partir de son compte Xikka.

Xikka offre ses services via une Plateforme Internet sécurisée.

1.2. Cibles

Xikka s'adresse à toute la population et à toutes les entreprises quel que soit leur taille, leur secteur d'activité.

- ✓ LES PARTICULIERS (personne physique capable d'effectuer des transactions financières.)
- ✓ LES ENTREPRISES (Secteur Privé dans tous les domaines d'activité et secteur public)
- ✓ LES ASSOCIATIONS A BUT NON LUCRATIF (Mutuelle, ONG etc.)

1.3. Organisation et fonctionnement

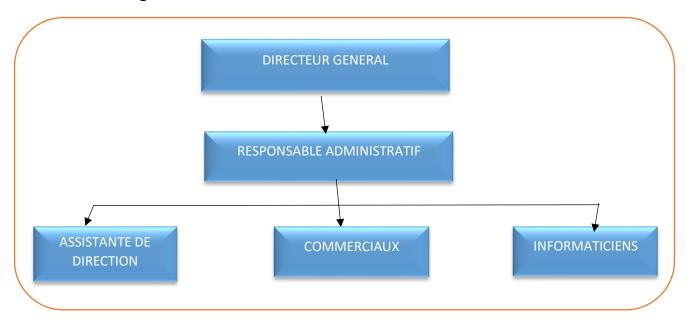


Figure 2- Organigramme Xikka

CHAPITRE II : CONTEXTE GENERAL DU PROJET

1. Problématique

Xikka dispose de plusieurs commerciaux qui sont chargés de faire connaitre l'entreprise et de gagner de nouveaux clients.

Après les prospections ou la recherche de nouveaux clients, les commerciaux de Xikka disposent des supports papiers pour l'enregistrement des informations des clients prospectés. Ces supports peuvent disparaitre facilement.

Lorsqu'un client pose une réclamation, les commerciaux de Xikka ont du mal à enregistrer cette réclamation.

L'absence d'un commercial peut être synonyme de perte de clients et de rendez-vous, rattaché a ce agent.

Pour pallier à ces différents problèmes, nous avons décidé de concevoir une application Web pour faciliter la tâche aux commerciaux de Xikka.

2. Présentation du projet

Notre application Web sera nommée 'GestClient'. L'application comportera trois types de profil à savoir :

Le profil administrateur

L'administrateur sera le gérant des deux autres utilisateurs. Il aura la possibilité de :

- ✓ Créer les profils des utilisateurs,
- ✓ Bloquer le profil d'un utilisateur,
- ✓ Consulter la liste des clients de chaque commercial,
- ✓ Consulter la liste de tous les clients.
- ✓ Consulter les rendez-vous de chaque commercial,
- ✓ Consulter les réclamations notées par chaque commercial,
- ✓ Imprimer la liste des clients de chaque commercial,
- ✓ Imprimer la liste des rendez-vous de chaque commercial,
- ✓ Imprimer la liste des réclamations notées par chaque commercial.

Le profil commercial

Le commercial sera l'utilisateur qui interagira plus avec l'application. Il aura la possibilité de :

- ✓ Enregistrer les contacts des clients prospectés,
- ✓ Enregistrer les rendez-vous pris avec les clients,
- ✓ Enregistrer les réclamations de ces clients,
- ✓ Envoyer des emails à ces clients,
- ✓ Reporter un rendez–vous avec le motif,
- ✓ Archiver ou annuler un rendez-vous avec le motif,
- ✓ Archiver ou annuler une réclamation avec le motif,
- ✓ Consulter ces différents rendez-vous pris avec ses clients,
- ✓ Consulter ces différents rendez-vous reportés de chaque client
- ✓ Consulter toutes les réclamations portées par ses clients,
- ✓ Avoir le nombre total de ces clients enregistrés dans l'application,
- ✓ Avoir le nombre total des réclamations portées par ces clients,
- ✓ Imprimer la liste de ces clients,
- ✓ Imprimer la liste de ces rendez-vous,
- ✓ Imprimer la liste de réclamations.

Le profil chef commercial

Comme le nom du profil l'indique, cet utilisateur sera le chef des commerciaux. Il fera les mêmes tâches que l'administrateur Cependant il n'aura pas la possibilité de créer et de bloquer un profil utilisateur Par contre il pourra noter chaque commercial en fonction de son travail.

3. Objectifs du projet

Les objectifs majeurs de 'GestClient' sont :

- ✓ Permettre à l'entreprise de suivre ses clients,
- ✓ Fidéliser ses clients.
- ✓ Suivre toutes les activités de chaque commercial.

CHAPITRE III:

CONCEPTION ET REALISATION DE L'APPLICATION

I. CONCEPTION

1. Diagramme des cas d'utilisations

Les diagrammes de cas d'utilisation sont des diagrammes UML(Unified Modeling Language en français Langage de Modélisation Unifié) utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils permettent de décrire l'interaction entre l'utilisateur (humain ou machine) et le système. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases). Voici présenté ci-dessous le diagramme des cas d'utilisation de notre application :

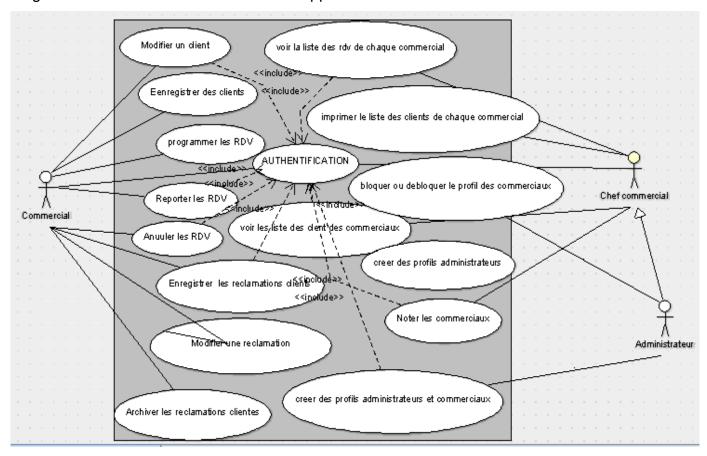


Figure 3- Diagramme des cas d'utilisations

1.1. Description des cas d'utilisations

✓ Authentification :

C'est la porte d'accès aux différentes fonctionnalités de l'application. L'application vérifie que l'utilisateur a le droit de se connecter. Le *mail* et le *mot de* passe de l'utilisateur sont vérifiés, s'ils sont corrects, l'accès est accordé et l'utilisateur accède aux fonctionnalités propices.

✓ Enregistrer Client :

L'utilisateur connecté peut enregistrer les clients dans le système.

✓ Programmer RDV

L'utilisateur connecté peut programmer des Rendez-vous avec ces clients dans l'application.

✓ Enregistrer réclamations Clientes

L'utilisateur connecté peut enregistrer des réclamations portées par des clients ou par ses clients dans l'application.

✓ Bloquer ou débloquer

L'utilisateur connecté peut bloquer ou débloquer le compte des commerciaux dans l'application. Fonctionnalité réservée uniquement à l'administrateur.

√ Reporter RDV

L'utilisateur connecté peut reporter des Rendez-vous programmés avec un client à une nouvelle date.

✓ Créer Profils administrateurs et commerciaux

L'utilisateur (uniquement un administrateur) peut créer les profils des commerciaux, des chefs commerciaux et d'autres profils administrateurs dans l'application.

✓ Imprimer

L'utilisateur connecté peut imprimer selon ses besoins des documents en PDF.

2. Diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. Ils donnent une vision des enchainements des activités propres

à une opération ou à un cas d'utilisation. Dans la suite, nous allons présenter de façon détailler les diagrammes d'activés de quelques cas d'utilisation :

2.1. Diagramme d'activité « Authentification »

Le cas d'utilisation " Authentification " est illustré par le diagramme d'activité suivant :

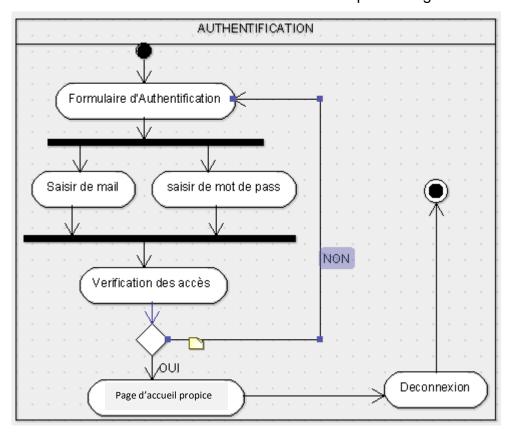


Figure 4- Diagramme d'activité « Authentification »

Description du diagramme

Lorsque l'utilisateur démarre l'application, le formulaire d'authentification lui est présenté. Il saisit son adresse mail et son mot de passe, le système tentera d'authentifier. Si le mail et le mot de passe d'utilisateur sont corrects, la page d'accueil propice à l'utilisateur s'affiche. Au cas où ses accès sont incorrects l'utilisateur serra amené à saisir à nouveau son mot de passe et son mail.

2.2. Diagramme d'activité « créer profils commerciaux et administrateurs »

Le cas d'utilisation " *Créer Profils commerciaux et Administrateurs* " est illustré par le diagramme d'activité suivant :

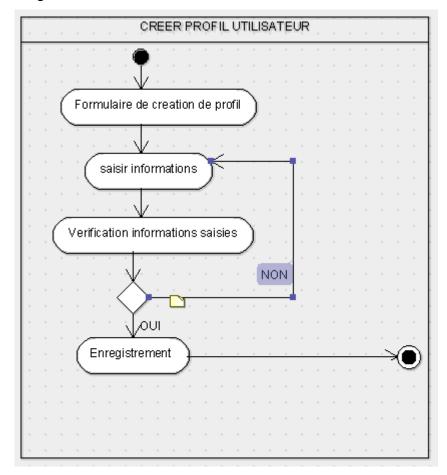


Figure 5- Diagramme d'activité « Créer Profil Utilisateur »

✓ Pré condition

L'utilisateur doit s'authentifier avant d'effectuer cette opération.

Description du diagramme

L'utilisateur demande la fenêtre de création de nouveau profil en cliquant sur le bouton 'nouvel utilisateur'. Le système affichera la fenêtre de saisie à l'utilisateur connecté. Après avoir finir de saisir les différentes informations relatives au nouvel utilisateur, l'utilisateur valide en cliquant sur le bouton 'enregistrer'. Le système va vérifier ses informations. Si ces informations sont conformes à celles d'un utilisateur existant dans l'application, alors le système n'enregistre pas ces informations, affiche un message d'erreur à l'utilisateur et réaffiche à nouveau le formulaire de saisie et

l'utilisateur pourra saisir à nouveau de nouvelles informations. Dans le cas contraire le profil de cet utilisateur est créé et le système affiche un message de succès à l'utilisateur.

2.3. Diagramme d'activité « Enregistrer des Clients »

Le cas d'utilisation " *Enregistrer des clients* " est illustré par le diagramme d'activité suivant :

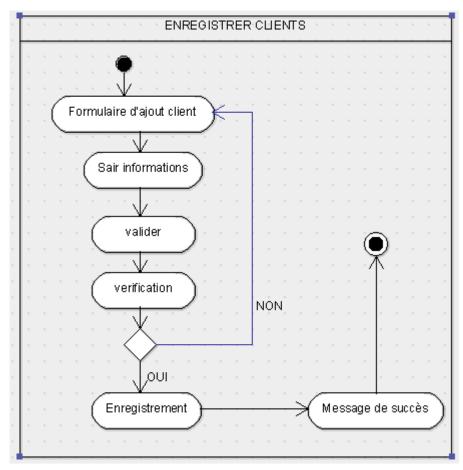


Figure 6- Diagramme d'activité « enregistrer des Clients »

✓ Pré condition

L'utilisateur doit s'authentifier avant d'effectuer cette opération.

Description du diagramme

L'utilisateur demande la fenêtre d'enregistrement d'un client en cliquant sur le bouton 'Client/Ajouter'. Le système affichera la fenêtre de saisie à l'utilisateur connecté. Après avoir finir de saisir les différentes informations relatives au nouveau client, l'utilisateur valide en cliquant sur le bouton 'enregistrer'. Le système va vérifier ses informations, si ces informations sont conformes à celles d'un client existant dans

l'application ou si elles ne sont pas correctes c'est- à- dire ne respectent pas les contraintes posées par l'application, alors le système n'enregistre pas ces informations. Affiche un message d'erreur à l'utilisateur et réaffiche à nouveau le formulaire de saisir et l'utilisateur pourra saisir à nouveau de nouvelles informations. Dans le cas contraire le client est enregistré dans l'application et un message de succès est affiché à l'utilisateur pour lui signifier la réussite de son action.

3. Diagramme des classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Ces objets sont représentés par des classes. Une classe est la description des comportements, et les responsabilités d'un ensemble d'objets qui ont des réactions communes. Les principales classes de notre diagramme des classes sont :

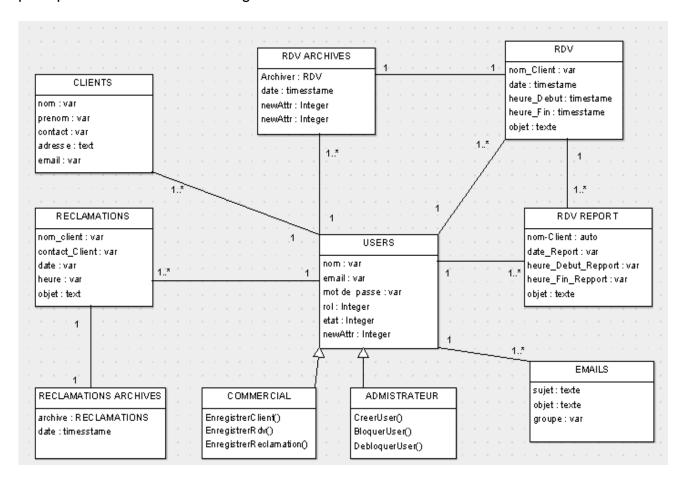


Figure 7- Diagramme des classes

3.1. Description du diagramme des classes

Ce diagramme présente les classes principales de notre système :

- ✓ USERS: Elle représente l'entité centrale de notre application web. Nous pouvons remarquer par sa relation avec toutes les autres entités du diagramme de classe. Cela signifie que toute personne, avant d'effectuer toute opération, doit être d'abord enregistrée comme utilisateur (commercial ou administrateur) dans l'application.
- ✓ CLIENTS : cette classe représente les clients enregistrés du système.
- ✓ RDV : cette classe représente les rendez-vous pris avec les clients.
- ✓ RDV REPORTS : cette classe représente les rendez-vous reportés par les commerciaux.
- ✓ RDV ARCHIVES: cette classe représente les rendez-vous archivés par les commerciaux.
- ✓ RECLAMATIONS: cette classe représente les réclamations portées par les clients.
- ✓ RECLAMATIONS ARCHIVES: cette classe représente les réclamations archivées par les commerciaux.
- ✓ EMAILS : cette classe représente les emails envoyés par les commerciaux aux différents clients.

II. PHASE DE REALISATION

1. Choix des technologies

1.1. Choix du langage de programmation

L'un des tournants les plus décisifs dans la réalisation d'un projet est le choix des outils de programmation. Il convient de choisir en considérant les besoins de l'entreprise, le délai de réalisation du projet pour éviter de devoir changer de langage en cours de projet, ce qui serait une perte de temps. D'où, un langage facile à apprendre permettra d'atteindre les objectifs fixés. Il existe cependant un grand nombre de langages informatiques avec lesquels il est possible de réaliser une application web.

Pour choisir un langage de programmation adéquat, il convient de comparer les langages.

Ce diagramme ci– dessous montre les pourcentages d'utilisation de top 5 de langages utilisés dans la programmation web en 2017. Selon les statistiques du site <u>w3techs.com</u>.

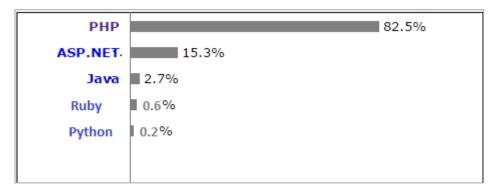


Figure 8- Langages de programmation (Source : <u>w3techs.com</u>)

Après analyse de l'histogramme ci-dessus, nous constatons que PHP est utilisé par 82.5% de tous les sites, ASP.NET 15.3%, java 2.7%, Ruby 0.6%, et Python 0.2%.

1.1.1. Comparaison des langages retenus

PYTHON

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par le programmeur GUIDO VAN ROSSUM.

Ce langage a été baptisé ainsi en hommage à la troupe de comiques les « Monty Python » (série télévisée).

Python est un langage de programmation interprété, c'est-à-dire que les instructions que vous lui envoyez sont « transcrites » en langage machine au fur et à mesure de leur lecture.

Son interpréteur fonctionne sur de nombreux systèmes d'exploitation tels que les systèmes linux, Microsoft Windows ou encore Mac OSX.

Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches. On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses extensions destinées au calcul numérique.

RUBY

Ruby est décrit comme un langage de script orienté objet. Il a pour but de combiner le meilleur des langages de programmation procéduraux et fonctionnels pour les adapter dans le monde des langages script.

Il est utilisé dans Apache pour générer des pages web et dans PostgreSQL où des commandes Ruby sont exécutées sur le serveur de base de données.

Son interpréteur fonctionne sur de nombreux systèmes d'exploitation tels que les systèmes linux, Microsoft Windows ou encore Mac OSX.

Ruby possède de nombreuses bibliothèques de fonctionnalités, appelés des gems, qui peuvent être adjointes au langage. Le langage possède également un gestionnaire de paquets appelé RubyGems qui permet d'installer ces gems. Parmi elles, on retrouve des bibliothèques pour permettre de communiquer avec le protocole POP3 pour l'envoi de mails, MySQL pour la gestion des bases de données. Il existe également des bibliothèques qui permettent de générer des documents PDF.

JAVA EE

Le terme « Java EE » signifie *Java Enterprise Edition*, et était anciennement raccourci en « J2EE ». Il fait quant à lui référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE, et elle y ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif émancipé de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

ASP.NET

ASP.NET est la plateforme de développement de Microsoft permettant la réalisation d'applications web. Il peut être utilisé avec n'importe quel langage de programmation pour la plateforme .NET (Visual Basic .NET, C#).Son outil de développement est Visual Studio.

PHP

Le langage PHP, qui signifie "PHP: Hypertext Processor", est un langage interprété (un langage de script). Il peut être exécuté par un serveur. Le serveur peut interpréter et générer le code PHP. Le langage PHP a été créé au début de l'automne 1994 par RASMUS LERDORF. Aujourd'hui, le langage PHP en est à sa septième version. La syntaxe du langage provient de celles du langage C, Perl et de Java. Il peut être utilisé sur les systèmes d'exploitation les plus répandus tels que Linux, Microsoft Windows ou Mas OSX. Il est également présent sur de nombreux serveurs web comme Apache ou IIS. Il permet de programmer de deux manières différentes, à savoir en procédural ou en orienté objet. Il supporte un grand nombre de bases de données entre autres MySQL, POSTGRES. Il est également possible de générer divers documents à la volée avec PHP, comme des images ou des fichiers PDF. Enfin, le langage permet de communiquer avec d'autres protocoles comme LDAP, IMAP, POP3 ou encore HTTP.

1.1.2. Conclusion partielle

A l'issue de notre comparaison, PHP est le langage retenu. En effet, il s'agit d'un langage facile d'apprentissage, accessible sur la plupart des systèmes d'exploitation et très populaire sur le web, ce qui permet un meilleur support et une meilleure maintenance. De plus, il s'agit d'un langage déjà éprouvé depuis plusieurs années et donc assez robuste pour le développement de notre application.

1.2. Choix du Framework PHP

Un framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant un « squelette » de programme. Il est conçu en vue d'aider les programmeurs dans leur travail. L'organisation du framework vise la productivité maximale du programmeur qui va l'utiliser, gage de temps, à réaliser un bon code, facile à maintenir. De plus, un Framework offre des briques prêtes à être utilisées, ce qui vous évite de réinventer la roue, et surtout qui vous permet d'utiliser des briques puissantes et éprouvées. D'où un outil indispensable pour la création de notre application web. Cependant, Il existe

une grande quantité de Framework PHP. Chacun présentant des avantages et des inconvénients, il va falloir choisir celui qui convient le plus à la réalisation de notre application. Pour ne pas failli dans notre choix, prenons un peu de recul et jetons un coup d'œil sur les tendances de 2017 dans le monde du Framework PHP.

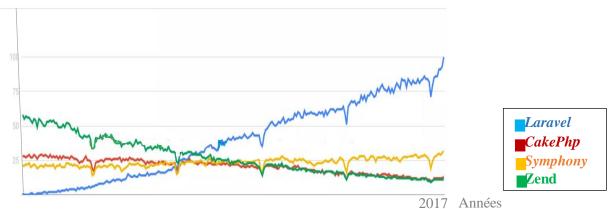


Figure 9- Frameworks PHP (Source : <u>engines.com</u>)

La figure ci-dessus présente les tendances des frameworks PHP les plus populaires : Laravel, Symphony, Zend et CakePhp. Nous allons nous pencher sur les 2 qui sont en tête de la liste actuellement c'est- à- dire Laravel et Symphony.

1.2.1. Comparaison des Frameworks retenus

SYMPHONY

Symphony est un Framework PHP français dont le créateur est FABIEN POTENCIER. Il est open source et très utilisé. Le framework symphony est très modulaire et il est possible d'en choisir uniquement certains morceaux , conçu pour s'intégrer facilement avec un ORM(Object Relational Mapping) tel que doctrine permettant de faciliter la persistance d'objets en base de données, gestion multi-langues simplifiée, système intégré d'améliorations des performances et forte capacité d'évolution.

LARAVEL

Laravel est un framework web open-source écrit en PHP, créé par TAYLOR OTWEL, il initie une nouvelle façon de concevoir un framework en utilisant ce qui existe

de mieux pour chaque fonctionnalité. Par exemple toute application web a besoin d'un système qui gère les requêtes HTTP. Plutôt que de réinventer quelque chose le concepteur de Laravel a tout simplement utilisé celui de Symphony en l'étendant pour créer un système de routage efficace. En quelque sorte OTWEL a fait son marché parmi toutes les bibliothèques disponibles. Il possède également un ensemble de composants qui lui sont propres :

- ✓ Un système de routage perfectionné. Laravel intègre en son sein un système de routage. Il permet de créer des chemins url qui sont simples et naturels pour accéder aux différentes fonctionnalités du site en développement Le routeur de Laravel permet de définir des URL créatives, associables à différentes zones de votre application simultanément.
- ✓ Un créateur de requêtes SQL et un ORM performants Un Mapping objetrelationnel (en anglais object-relational Mapping ou ORM) est une technique de programmation informatique qui définit des correspondances entre une base de données et les objets du langage utilisé L'utilisation de la programmation orientée objet avec une base de données relationnelle nécessite de convertir les données relationnelles en objets et vice-versa. Ceci conduit à programmer cette conversion pour chaque objet et donc à dupliquer énormément de code similaire. C'est donc principalement à ce problème que tente de répondre un ORM. Laravel comporte en son sein un ORM performant qui s'appelle Eloquent. Pour mapper convenablement les données de la base de données et les objets implémentés sous Laravel, il faut utiliser ce qu'on appelle le CRUD. CRUD (Create, Read, Update, Delete) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données. Concrètement, il s'agit de gérer et d'interroger la base de données directement avec Laravel, dans du code PHP. Un fichier de configuration permet de paramétrer la connexion à une base de données. c'est tout ce qu'il y'a à faire. Laravel s'occupera alors de faire ce que vous lui direz.
- ✓ Un moteur de Template efficace(Blade). Blade est le moteur de Template proposé par Laravel. Il est simple à utiliser et puissant. Il donne la

possibilité d'utiliser des éléments communs à plusieurs pages (par exemple : header, menu, footer etc.) en agissant sur un fichier uniquement, grâce à un système d'héritage. Cela dans le but de réduire la redondance du code et d'améliorer la visibilité, la compréhension et la maintenabilité du code. Blade apporte également dans son panel d'outil des tests de condition et de boucle, grâce à un langage dédié.). En Bref, Blade permet de faciliter la conception des différentes pages et de leurs composants.

- ✓ Un système d'authentification pour les connexions. Laravel rend la mise en œuvre d'un module d'authentification beaucoup plus simple. Un fichier de configuration en son sein permet de le paramétrer grâce à plusieurs options pour personnaliser le comportement de l'authentification. Il prend notamment en charge la gestion des cookies et de la session. Aussi, il offre la possibilité de définir de quelle façon récupérer les différents utilisateurs enregistrés dans la base de données. Il est compatible et interopérable avec l'ORM Eloquent dont nous avons parlé avant qui nous servira ici à récupérer et gérer les utilisateurs.
- ✓ Un système de validation. Laravel fournit plusieurs approches différentes pour valider les données entrant de votre application provenant de formulaire par exemple. Le contrôle des valeurs qui sont transmises entre les pages est alors facilité. En effet, plus besoin de coder en dur cette analyse de conformité des données, qui peut être un processus lourd, rébarbatif, et en général non exempts de défaut de conception. Laravel propose des méthodes robustes améliorant ce travail.
- ✓ Un système de pagination. Dans d'autres cadres, la pagination peut être très douloureuse. Laravel fait un jeu d'enfant grâce à sa simple méthode *paginate*.
- ✓ Un système de migration pour les bases de données. Laravel propose un outil permettant de créer et de mettre à jour un schéma de base de données. Toujours dans le même esprit qu'Eloquent, il s'agit de centraliser toute la gestion de la base de données au sein de Laravel. Cet outil permet de

- prendre en charge tout ce qui concerne la maintenance des tables d'une base de données (création de tables, de colonnes, d'index, etc.).
- ✓ Un système d'envoi d'emails. Laravel propose des fonctionnalités très pratiques permettant d'envoyer des emails, grâce à l'outil SwiftMailer, riche en fonctionnalités. Encore une fois, dans l'esprit de centraliser toutes les fonctionnalités d'un site au sein d'un même outil, la construction des vues d'un email est similaire en tout point à la construction des vues d'une page du site. Il suffit pour cela d'utiliser Blade, le moteur de Template.
- ✓ Un système d'autorisations. Laravel contient un système (*middleware*) qui permet de contrôler l'accès des utilisateurs aux différentes pages d'une application.
- ✓ Une gestion des sessions. Les sessions sont utilisées pour stocker des informations sur l'utilisateur au fur et à mesure de la navigation. Laravel fournit de nombreux pilotes pour gérer les données de session. Par défaut, le pilote de fichier est utilisé parce qu'il est léger.

1.2.2. Conclusion partielle

Parmi ces 2 frameworks PHP, tous présentent des avantages et des inconvénients certains. Mais notre choix s'est porté sur Laravel. En effet, celui-ci est un Framework PHP très populaire en entreprise. Il présente également un ORM très puissant, appelé Eloquent, qui permet de récupérer les éléments présents en base de données de manière plus rapide et efficace des requêtes optimisées pour la base de données. Laravel possède également un système de migration de base de données. Et en plus des mêmes atouts que Symphony, il propose une orchestration qui le différencie des autre frameworks PHP. Il va permettre ainsi de développer et déployer plus rapidement notre application.

1.3. Choix du langage de Modélisation : UML

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML). Un langage de modélisation graphique à base de diagrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en objet. Vus ces nombreux atouts, nous l'avons rapidement choisi comme langage de modélisation de notre application.

1.4. Architecture: MVC

Le choix de l'architecture (l'organisation) de notre code s'est très vite porté sur le Pattern MVC. En effet il est un patron de conception (design pattern en anglais) très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données. Il a été massivement adopté comme modèle pour la création d'applications web (dites « client léger »). Il faut noter que MVC n'est pas associé à un langage de programmation et qu'il peut être utilisé avec plusieurs langages. MVC est beaucoup utilisé avec PHP par exemple ou même JAVAEE. En général MVC agit comme une couche supplémentaire qui vient se greffer par-dessus un langage.

Le **M** de MVC signifie **Modèle**. Il s'agit des données et de l'état de votre application web. Il représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données, garantit leur intégrité et offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur).

Le **V** de MVC signifie la **Vue**, il traite de ce qu'on voit à l'écran dans le navigateur web. Il s'agira globalement de code HTML et de CSS. Le but de la vue est de présenter les données issues du modèle mais sans les modifier, sans en interpréter le contenu.

Le **C** de MVC signifie **Contrôleur**. Il fait le lien entre la vue et le modèle. Le contrôleur gère les interactions avec l'utilisateur et détermine quels traitements doivent être effectués pour une action donnée. D'une manière générale, il va utiliser les données du modèle, les traiter en fonction de l'action de l'utilisateur, et les envoyer à la vue afin qu'elle les affiche.

Nous verrons que le contrôleur est au cœur du système, il interprète et « contrôle » les données venant de l'utilisateur, comme des données venant d'un formulaire ou bien simplement une action faite via une URL. MVC est donc une bonne solution pour réaliser des applications web.

1.5. Choix du système de gestion de bases de données (SGBD)

La question de la base de données est un élément très important avant même la phase de programmation d'une application. Car c'est la base de données qui va stocker toutes nos données qui seront manipulées dans notre application. D'où un choix judicieux nous évitera beaucoup de soucis à savoir la limite des enregistrements. Il convient donc de choisir encore une fois en fonction du besoin, mais aussi des contraintes de maintenabilité et des critères de performance un bon système de gestion de base de données. Pour cela, une première étape consiste à étudier la popularité des solutions disponibles.

	Oct. 2017	Novembre 2016	SGBD
1.	1.	1.	Oracle 🔠
2.	2.	2.	MySQL 🔠
3.	3.	3.	Microsoft SQL Server 🖽
4.	4.	4.	PostgreSQL 🔠
5	5	5	MongoDB 🔠

Figure 10- Classement de popularité des SGBD (Source : <u>db-engines.com</u>)

La figure ci-dessus représente le classement des 5 systèmes de gestion de base de données les plus utilisés. On retrouve ainsi Oracle en tête de liste, suivi de MySQL, Microsoft SQL Server, PostgreSQL et MongoDB. En vue de choisir le meilleur pour notre application, nous allons établir une comparaison entre ces différents SGBD.

1.5.1. Présentation des SGBD

ORACLE

Oracle database est un système de gestion de base de données (SGBD) fourni par Oracle Corporation. C'est l'un des SGBD les plus populaires au niveau mondial. Il fonctionne sur un système relationnel et même sur un modèle relationnel objet depuis la version 8.

Ce système propriétaire a été développé notamment par LARRY ELLISON, le cofondateur de l'entreprise Oracle. Il est implémenté en C et C++ et est disponible sur la plupart des systèmes d'exploitation (Linux, OSX, Windows). Il est également compatible avec une grande variété de langage de programmation, dont PHP.

Oracle est un SGBD permettant d'assurer :

- ✓ La définition et la manipulation des données
- ✓ La cohérence des données
- ✓ La confidentialité des données
- ✓ L'intégrité des données
- ✓ La sauvegarde et la restauration des données

MYSQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) développé également par Oracle Corporation. Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde. Il s'agit d'une base de données relationnelle SQL, développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur. C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante. MySQL fonctionne sur de nombreux systèmes d'exploitation différents, IBM, Linux, Mac OS, Windows...

Un SGBD puissant dans la manipulation d'une base de données comportant des données de petites tailles.

MICROSOFT SQL SERVER

Microsoft SQL Server est un système de gestion de base de données développé par Microsoft et sorti en 1989. Disponible sous licence commerciale, une licence gratuite est également disponible mais est limité en termes de fonctionnalités. Actuellement dans sa version 2014, Microsoft SQL Server est développé en C++ et supporte moins de langage qu'Oracle, MySQL et POSTGRES. Il fonctionne sous les systèmes OS Windows et Linux (depuis mars 2016).

POSTGRESQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MySQL), ou propriétaires (comme Oracle, Microsoft SQL Server).

PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises. C'est un type de base de données très utile lorsque les données stockées sont très complexes et nombreuses. PostgreSQL peut stocker plus de types de données que les types simples traditionnels entiers, caractères, etc. Il est largement reconnu pour son comportement stable, proche d'Oracle. PostgreSQL fonctionne sur de nombreux systèmes d'exploitation différents, IBM, Linux, Mac OS, Windows et Supporte le langage PHP grâce à la librairie « pgsql » qu'intègre PHP.

MONGODB

MongoDB (de l'anglais humongous qui peut être traduit par «énorme») est un système de gestion de base de données orientée documents, à la différence de ceux présentés ci-dessus. Il est écrit en C++. Cette tendance vise à gérer les bases de données de manière transparente, sans avoir recours, comme son nom l'indique, au SQL classique, dans un souci de simplicité. Ce sont des bases de données prévues pour des applications dites Big Data. Les données prennent la forme de documents enregistrés eux-mêmes dans des collections, une collection contenant un nombre quelconque de documents. Les collections sont comparables aux tables, et les documents aux enregistrements des bases de données relationnelles. Contrairement aux bases de données relationnelles, les champs d'un enregistrement sont libres et peuvent être différents d'un enregistrement à un autre au sein d'une même collection. Le seul champ commun et obligatoire est le champ de clé principale ("id"). MongoDB ne permet ni les requêtes très complexes standardisées, ni les JOIN, mais permet de programmer des requêtes spécifiques en JavaScript pour des données mises en page il supporte une grande variété de langage modernes et

anciens, dont entre autres PHP, et est disponible sous Linux, OSX, Windows et Solaris.

1.5.2. Conclusion partielle

Après présentation de ces différents SGBD, deux SGBD ont retenu notre attention : MySQL et PostgreSQL. Dans les deux cas, il s'agit des SGBD très populaires, libre et très répandus. Bien vrai que MySQL est plus populaire que PostgreSQL, notre choix c'est porté sur PostgreSQL. Car c'est un SGBD qui est plus adapté dans la manipulation d'une base de données qui regorge des données plus nombreuses par rapport à MySQL.

2. Architecture et fonctionnement de l'application

La réussite d'un projet dans le délai fixé dépend de l'organisation et de la démarche à suivre des réalisateurs. C'est la raison pour laquelle nous avons choisi le Framework Laravel qui nous oblique à bien structurer notre code selon le model MVC. Dans ce paragraphe, nous allons présenter l'architecture de Laravel et son impact dans développement de notre application.

2.1. Organisation et fonctionnement du Framework Laravel

Le schéma ci-dessous présente les différents dossiers que compose un projet Laravel et leur organisation :

] арр	11/11/2017 10:37	Dossier de fichiers
bootstrap	11/11/2017 10:37	Dossier de fichiers
📗 config	11/11/2017 10:37	Dossier de fichiers
📗 database	11/11/2017 10:37	Dossier de fichiers
ll public	11/11/2017 10:49	Dossier de fichiers
ll resources	11/11/2017 10:38	Dossier de fichiers
ll routes	11/11/2017 10:38	Dossier de fichiers
📗 storage	11/11/2017 10:38	Dossier de fichiers
ll tests	11/11/2017 10:38	Dossier de fichiers
📗 vendor	11/11/2017 10:39	Dossier de fichiers
.env	04/11/2017 09:02	Fichier ENV

Figure 11- Structure des fichiers sous Laravel

2.1.1. Description des dossiers

Soulignons que nous avons installé la version 5.5 de laravel. Cette version nécessite au moins la version 5.5.9 de PHP.

✓ app

Le dossier **app** est le dossier le plus important de notre projet. C'est notre terrain de jeu. C'est celui qui contient notre application c'est à dire, tous nos modèles et nos contrôleurs (schéma **MVC**).

✓ boostrap

Ce dossier contient des fichiers qui gèrent le démarrage de Laravel.

✓ config

Le dossier config contient les fichiers de configuration du Framework. Chaque fichier correspond à une fonctionnalité configurable comme la gestion de la base de données avec le fichier database.php ou le fichier mail.php pour l'envoi des mails. D'où la configuration notre base de données est géré dans ce dossier.

✓ database

Le dossier database contient principalement les migrations. Les migrations sont des fichiers permettant de décrire notre base de données afin de permettre à Laravel de créer, modifier ou supprimer les tables et les colonnes automatiquement pour nous.

✓ public

Le dossier public contient tous les fichiers accessibles directement par nos utilisateurs comme les images, les feuilles de style ou encore les fichiers JavaScript. Il contient un fichier index.php qui est la porte d'entrée de notre application. C'est le seul fichier PHP accessible de l'extérieur et il sera responsable de lancer le Framework et d'appeler notre code situé dans le dossier **app** pour le démarrage de notre application web.

✓ ressources

Le dossier contient un sous dossier views. Le dossier resources /views contient les vues (schéma MVC) de notre application. Les vues sont des fichiers majoritairement composés de HTML et sont chargés de la partie affichage de

notre application. Ce sont les fichiers destinés à présenter la réponse à l'utilisateur. C'est l'un des dossiers les plus importants après le dossier **app**.

✓ routes

Dans une architecture MVC dans le cas du Framework Laravel, toutes les requêtes des utilisateurs arrivent via le fichier **public**/index.php il est donc nécessaire de faire le lien entre l'URL entrée par le visiteur et le code qui sera exécuté. C'est dans le fichier **routes**/web.php que nous configurons ce lien. D'où ce dossier contient les URL de notre application.

√ storage

C'est le dossier de stockage d'informations temporaires diverses : sessions, logs, vues compilées, caches...

√ tests

Ce dossier est réservé par laravel pour des tests unitaires à lancer avec la commande artisan.

3. Conclusion partielle

Cette phase d'étude du projet nous a permis de faire le choix de notre langage de programmation, PHP, notre langage de modélisation, UML et le Framework PHP, LARAVEL. Soulignons que ces outils nous ont permis d'aller plus vite lors de la mise en place de notre application.

CHAPITRE IV:

PRESENTATION DE L'APPLICATION OBTENUE

1. Les interfaces

Dans cette partie, nous allons présenter quelques interfaces de notre application.

Authentification

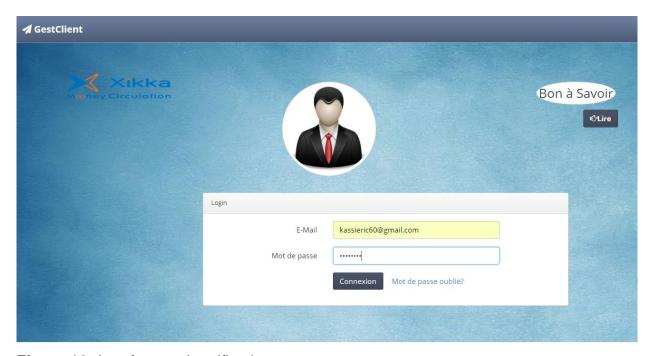


Figure 12- Interface authentification

Cette interface permet à l'utilisateur de se connecter à l'application. L'utilisateur doit entrer son email et son mot de passe pour accéder à l'application. En cas d'échec un message d'erreur s'affiche. Sinon, l'utilisateur accède à la page d'accueil de l'application selon son profil. Sur cette interface, l'utilisateur pourra avoir un autre mot de passe par mail s'il a oublié son mot de passe en cliquant sur le lien *mot de passe oublié* ?

Après la connexion, la page d'accueil s'affiche. Si l'utilisateur est un commercial, cette page d'accueil contient le nombre total de clients, le nombre de rendez-vous programmé, le nombre de réclamations portées par ces clients (*figure 13*).



Figure 13- Page accueil commercial

La page d'accueil présente un tableau contenant la liste des commerciaux avec les buttons généraux de l'application (*figure 14*) si l'utilisateur est un administrateur.

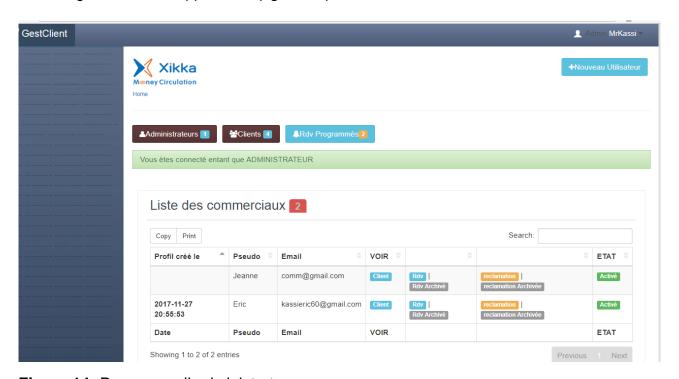


Figure 14- Page accueil administrateur

Modification de mot de passe et déconnexion

Lorsque l'utilisateur connecté clique sur son nom, en haut à droite de la page d'accueil, il a la possibilité de modifier son mot de passe et de se déconnecter. Voir les figures ci-dessous :



Figure 13.1- Modification Mot de mot

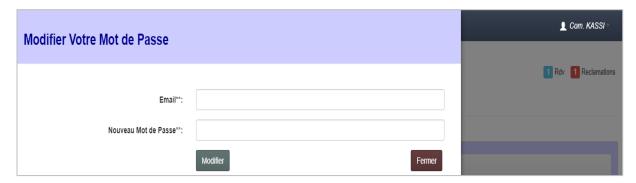


Figure 13.2-Interface de Modification Mot de Passe

Enregistrer Client

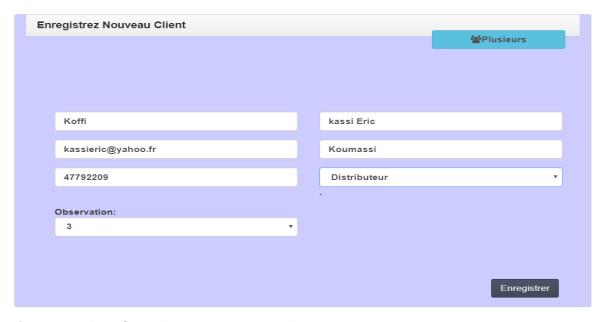


Figure 15- Interface d'enregistrement client

Cette interface permet au commercial connecté d'enregistrer ses clients prospectés dans l'application. Pour le faire, il renseigne ses champs puis valide en cliquant sur le bouton « *Enregistrer* ». Si le client existe préalablement dans le système, l'application affiche une alerte à l'utilisateur en lui signifiant à quel commercial ce client appartient.

En cliquant sur le bouton « *plusieurs* », le commercial est capable d'enregistrer plusieurs clients en un seul clique (*figure 16 et figure 17*).

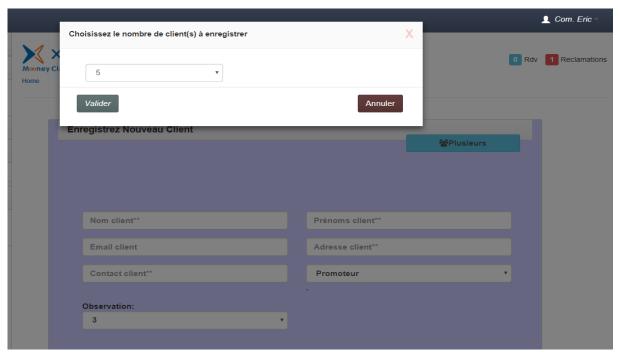


Figure 16- Interface nombre d'enregistrement plusieurs clients

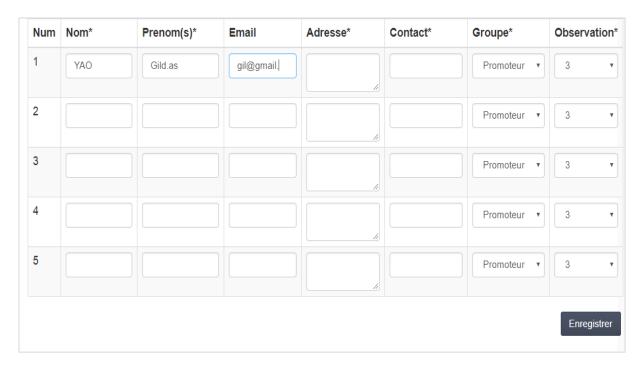


Figure 17- Interface d'enregistrement plusieurs clients

Liste des Clients

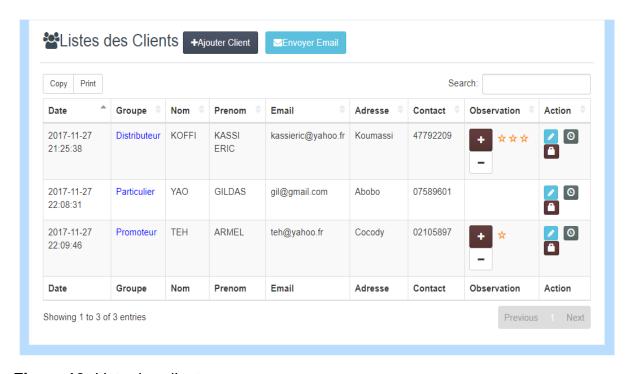


Figure 18- Liste des clients

Ce tableau présente la liste des clients du commercial connecté. Il a la possibilité de faire une recherche rapide. Il peut imprimer la liste de tous ces clients ou

des clients à une date précise. A partir de ce même tableau, l'utilisateur peut envoyer des emails à ses clients en cliquant sur le bouton « *Envoyer Email* » (*figure 19*).

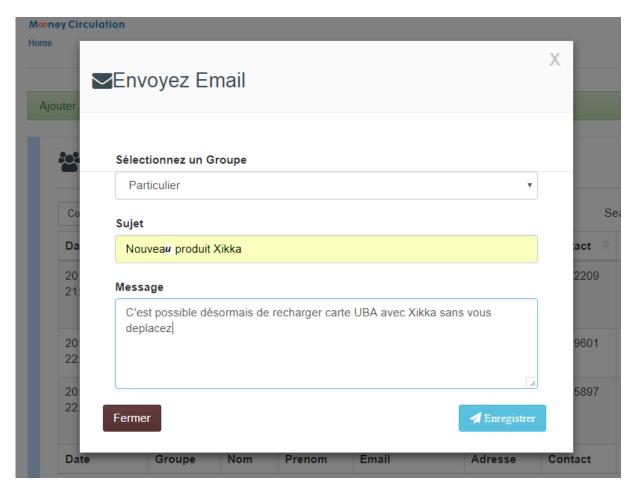


Figure 19- Interface d'envoi Email

Programmer RDV

Pénom:			
KASSI ERIC			
Adresse:			
Koumassi			
Date du RDV **:(m/j/a)			
11/30/2017			
Heure de Fin RDV **:			
12:00			
Hour Minute			
0 1 2 3 4 5 00 05 10			
6 7 8 9 10 11 15 20 25 12 13 14 15 16 17 30 35 40 18 19 20 21 22 23 45 50 55 Programmer RDV			

Figure 20- Interface RDV

Cette interface permet à l'utilisateur de programmer un rendez-vous avec un client choisi dans le tableau contenant la liste de ses clients. Il remplir les différentes informations puis clique sur le bouton « *Programmer RDV* » pour programmer un rendez-vous avec le client en question.

Reporter RDV

Pour reporter un rendez-vous, l'utilisateur se rend dans la liste des rendez-vous programmés. Sélectionne le rendez-vous en question pour le reporter en renseignant le motif du report. Voir la figure ci-dessous :

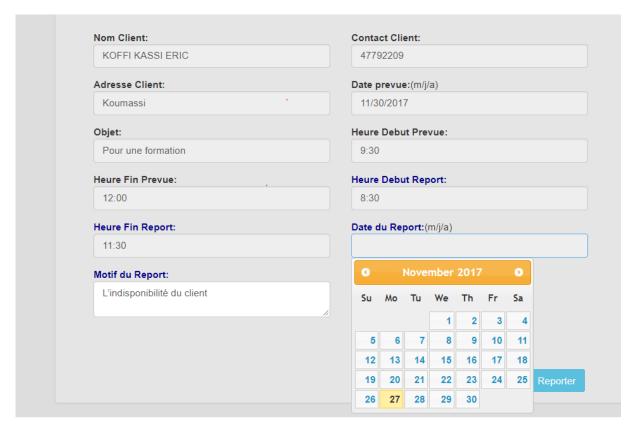


Figure 21- Interface reporté RDV

Liste des Rendez-vous programmés

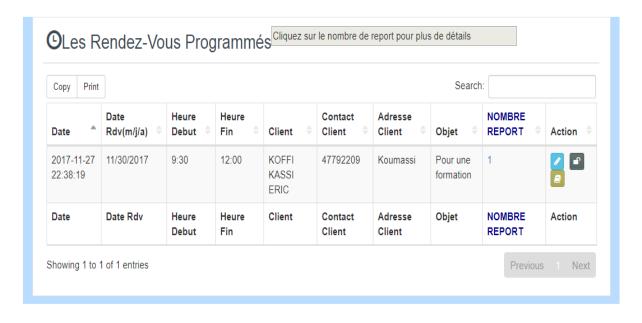


Figure 22- Liste des RDV

Cette liste contient, le nombre de report de chaque rendez-vous. Lorsque l'utilisateur clique sur le nombre de report, l'application lui affiche tous les détails possibles (*Figure 23*).

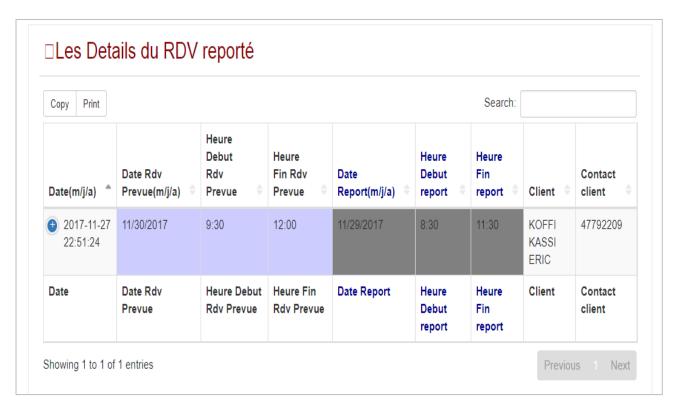


Figure 23- Détails RDV reporté

Création d'un profil utilisateur

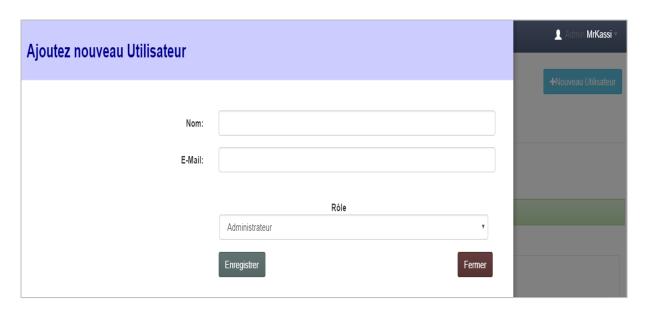


Figure 24- Interface de création profil utilisateur

Le profil des utilisateurs est créé uniquement par un administrateur. Lorsque ce dernier est connecté. Il clique sur le bouton « Nouvel Utilisateur » puis saisi les informations. Après validation des informations, l'utilisateur crée reçoit un mail contenant ses accès d'authentification à l'application.

Voir les clients de chaque commercial

Pour voir la liste des clients d'un commercial, l'administrateur clique devant le nom du commercial sur le bouton « client ». Et l'application affiche la liste des clients enregistrés par ce commercial (*figure 25*).

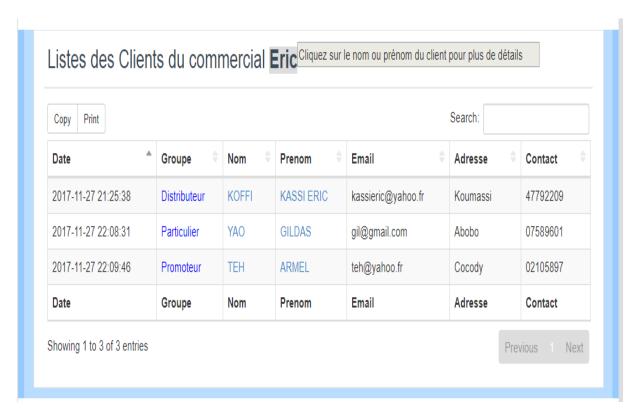


Figure 25- Liste des clients d'un commercial choisi

Lorsque l'administrateur clique sur le nom ou prénoms d'un client, nous avons plus d'information sur ce client à savoir les rendez- vous pris avec ce client et ces réclamations (*figure 26*).

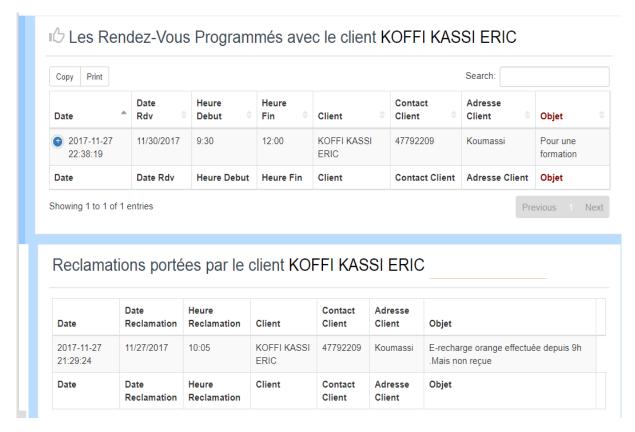


Figure 26- Détails client

Bloquer ou débloquer d'un profil commercial

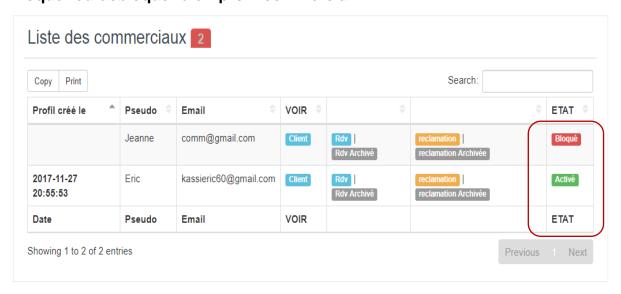


Figure 27- Interface bloquer/débloquer utilisateur

C'est seulement l'administrateur qui a la possibilité de bloquer ou débloquer le profil d'un commercial. Pour le faire, l'administrateur clique sur le bouton « *activé* » ou « *bloqué* » devant le commercial en question.

2. La sécurité de l'application

Il est primordial de nos jours de développer son application Web dans une logique de sécurité, dès le départ et tout au long du processus afin de se mettre à l'abri de quelques attaques. Assurer la sécurité d'une application Web est un métier à part entière et parfois difficile. Laravel nous offre :

- ✓ Un système d'autorisations bien pratique. Cela nous permet Dans le code de vérifier si la personne est authentifiée et quel est son degré d'habilitation.
- ✓ Le hashage. Laravel possède la façade Hash qui permet de faire du hashage avec Bcrypt. C'est ce qui est utilisé pour l'encryptions du mot de passe.
- ✓ Les injections SQL. Une injection SQL est une attaque qui consiste à ajouter à une requête inoffensive un complément qui peut être dangereux pour l'application. Grace à l'ORM Eloquent de laravel notre application est à l'abri automatiquement contre toutes injections SQL.

CONCLUSION

Le but de ce projet est la conception et la réalisation d'une application web de gestion des activités des commerciaux dans une Startup : cas de l'entreprise Xikka. Cette application a pour objectif principal : le suivi des clients et les activités des commerciaux de l'entreprise Xikka. Elle a été réalisée lors d'un stage effectué au sein de cette entreprise. Ce stage m'a ainsi permis de mettre en pratique les connaissances théoriques et pratiques acquises lors de ma formation, et d'en acquérir de nouvelles relatives à l'utilisation du Framework Laravel et du logiciel PostgreSQL. Bien que la plupart des fonctionnalités majeures de notre application soient achevées, certaines restent encore à faire notamment celles du profil chef commercial. Le bilan de ce stage est dans l'ensemble positif, les principaux buts du projet étant accomplis. La plus grande partie du travail qui m'a été demandée a été réalisée. Ce stage est une expérience enrichissante, il me permet de clore ma deuxième année de Master MIAGE (Méthodes Informatique Appliquée à Gestion et à l'Economie) et me donne également un aperçu du travail dans le milieu de la programmation.

RÉFÉRENCES BIBLIOGRAPHIQUES

Voici la liste des sources utilisées pour effectuer ce travail.

• LANGAGES DE PROGRAMMATION :

w3techs.com, consulté le 20 Octobre 2017.

PHP:

François Xavier Bois. *PHP 5 le guide complet*. Micro Application, 2007 <u>http://php.net</u>, consulté le 20 Octobre 2017.

RUBY:

<u>https://www.ruby-lang.org/fr/,</u> consulté le 20 Octobre 2017. <u>https://fr.wikipedia.org/wiki/Ruby,</u> consulté le 20 Octobre 2017.

JAVAEE:

javaee.github.io /tutoriel, consulté le 22 Octobre 2017. github.com /javaee, consulté le 22 Octobre 2017.

MVC :

Openclassroom.com/MVC, consulté le 20 Octobre 2017.

• FRAMEWORKS PHP:

<u>https://www.sitepoint.com/the-state-of-php-mvc-frameworks-in-2017</u>, consulté le 25 Octobre 2017.

<u>https://openclassrooms.com/courses/decouvrez-le-framework-php,</u> consulté le 25 Octobre 2017.

LARAVEL:

<u>https://laravel.com,</u> consulté le 10 Octobre 2017. <u>https://en.wikipedia.org/wiki/Laravel,</u> consulté le10 Octobre 2017. <u>https://openclassrooms.com/courses/decouvrez-le-framework-php-laravel,</u> consulté le10 Octobre 2017.

SYMPHONY:

https://Symphony.com, consulté le10 Octobre 2017.

https://openclassrooms.com/courses/developpez-votre-site-web-avec-le-

frameworkSymphony2/Symphony2-un-framework-php, consulté le10 Octobre 2017.

CAKE PHP:

<u>http://cakephp.org.</u>consulté le 12 Octobre 2017. <u>https://en.wikipedia.org/wiki/CakePHP,</u> consulté le 12 Octobre 2017.

https://www.grafikart.fr/formations/cakephp, consulté le10 Octobre 2017.

ZEND:

https://github.com/zendframework/zf2, consulté le10 Octobre 2017.

http://framework.zend.com, consulté le 12 Octobre 2017.

• COMPARAISON BASES DE DONNEES :

<u>https://www.wikivs.com/wiki/MySQL_vs_PostgreSQL</u>, consulté le 05 Novembre 2017.
<u>http://www.sitepoint.com/mysql-compared-with-postgresql/</u>, consulté le 05 Novembre 2017.
<u>http://www.db-engnines.com</u>, consulté le 05 Novembre 2017.

ORACLE DATABASE:

<u>https://fr.wikipedia.org/wiki/Oracle_Database,</u> consulté le 05 Novembre 2017. http://docs.oracle.com/database/121/index.htm, consulté le 05 Novembre 2017

MYSQL:

<u>https://fr.wikipedia.org/wiki/MySQL</u>, consulté le 05 Novembre 2017 <u>https://www.mysql.fr</u>, consulté le 05 Novembre 2017

POSTGRES:

<u>http://www.postgresql.org</u>, consulté le 05 Novembre 2017. <u>https://fr.wikipedia.org/wiki/PostgreSQL</u>, consulté le 05 Novembre 2017

MONGO DB:

<u>https://www.mongodb.org,</u> consulté le 05 Novembre 2017. <u>https://fr.wikipedia.org/wiki/MongoDB,</u> consulté le 05 Novembre 2017.

UML:

UML", "rapporte d'étude : Working paper de l'IDHEAP no 14/2002 nov. 2002 [en ligne] ", http://www.idheap.ch/, consulté le 10 Novembre 2017.

<u>http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activités,</u>consulté le10 Novembre 2017.

TABLE DES MATIERES

DEDICACE	ES	i
REMERCI	EMENTS	ii
LISTE DES	S FIGURES	iv
LISTE DES	S SYMBOLES	v
RESUME		vii
	T	
	CTION	
	E I : PRESENTATION GENERAL DE L'ENVIRONNEMENT	
	ntreprise	
	Services	
	Organisation et fonctionnement	
	E II : CONTEXTE GENERAL DU PROJET	
	blématiqueblématique	
	sentation du projet	
•	ectifs du projet	
	E III : CONCEPTION ET REALISATION DE L'APPLICATION	
	NCEPTION	
	gramme des cas d'utilisations	
	Description des cas d'utilisations	
	gramme d'activité	
2.1.	Diagramme d'activité « Authentification »	12
2.2.	Diagramme d'activité « créer profils commerciaux et administrateurs »	· 13
2.3.	Diagramme d'activité « Enregistrer des Clients »	14
3. Dia	gramme des classes	15
3.1.	Description du diagramme des classes	16
II. PHAS	E DE REALISATION	16
1. Cho	oix des technologies	16
1.1.	Choix du langage de programmation	16
1.2.	Choix du Framework PHP	19
1.3.	Choix du langage de Modélisation : UML	24
1.4.	Architecture : MVC	24

	1.5.	Choix du système de gestion de bases de données (SGBD)	25
2.	Arc	chitecture et fonctionnement de l'application	28
2	2.1.	Organisation et fonctionnement du Framework Laravel	28
3.	Co	nclusion partielle	30
CHA	PITR	E IV : PRESENTATION DE L'APPLICATION OBTENUE	31
1.	Les	s interfaces	32
2.	La	sécurité de l'application	43
CONCLUSION		44	
RÉFÉ	ÉREN	ICES BIBLIOGRAPHIQUES	45