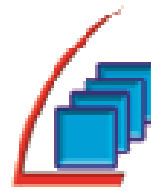


RÉPUBLIQUE TOGOLAISE
Travail-Liberté-Patrie

Ministère de la planification du
Développement



**Institut Africain d'Informatique,
Représentation du TOGO (IAI-TOGO)**
Rue de la Kozah Nyekonakpoè
07 BP : 12456 Lomé 07, Togo
Tél : +228 22 20 47 00/+228 22 21 27 06
E-mail : iaitogo@iaitogo.com
Site web : www.iai-togo.com



GESURI Sarl

Gestion Suppléance Recherches Intermédiation
Cabinet privé de management
545 Rue Koumoré,
en face de CCT-BATIMAT, (Lomé)
Tél: +228 22217299/ 22343421
E-mail: gesuri@cabinetgesuri.com
Site web: www.gesuriconseils.com

**PROJET DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME
D'INGENIEUR DES TRAVAUX INFORMATIQUES
OPTION : GENIE LOGICIEL**

Thème

***Gestion des clients, des créances et des discussions
instantanées dans un cabinet de conseil en management***

Période : 17 Juillet 2017 au 16 Octobre 2017

Rédigé et soutenu par :

ADUAYOM MESSAN Ekoue Enoc

Étudiant en Troisième Année

Année académique 2016-2017

Superviseur :
M. AKOUNDA Gbamra
Enseignant à IAI-TOGO

Maître de stage :
DELEGBE Gauss
Consultant à GESURI SARL



DEDICACES

Ce Mémoire est dédié spécialement à mes Parents pour les efforts qu'ils ont fournis pour ma formation et mon épanouissement ; à tous ceux qui de près ou de loin, par leur amour et leur soutien sont restés à mes côtés durant toutes ces années de formation.

Ici, je traduis mes plus profonds sentiments de reconnaissance à votre égard.

REMERCIEMENTS

Nos remerciements vont d'abord à DIEU tout puissant pour la santé et l'intelligence qu'il m'a conférées et pour toute son assistance dans mon travail.

Nous remercions tous ceux qui, de près ou de loin, nous ont aidé ou soutenu d'une manière ou d'une autre pour l'élaboration de ce Mémoire. Notre gratitude va particulièrement à :

- ☞ M. Kodjo AGBETI, Directeur Général de l'IAI-TOGO pour ses conseils ;
- ☞ M. Kossi N. AMEYIKPO Directeur des affaires académiques et scolaires de l'IAI-TOGO pour ses conseils ;
- ☞ M. AKPAKI Siméon, Directeur Général GESURI Sarl qui a bien voulu m'accorder ce stage ;
- ☞ M. DELEGBE Gauss, mon maître de stage qui a su me donner les bonnes directives pour répondre au mieux aux besoins de GESURI Sarl ;
- ☞ M. AKOUNDA, qui par sa disponibilité, ses idées, et ses conseils a supervisé notre stage et encadré mon travail. Nous avons admiré sa disponibilité, ses idées, et ses conseils ;
- ☞ A tous les professeurs de l'IAI-TOGO pour la qualité de leur formation ;
- ☞ A tous nos camarades de promotion (2014-2017) ;
- ☞ Aux membres du jury qui ont bien voulu apprécier notre travail.

AVANT-PROPOS

L'Institut Africain d'Informatique, représentation du TOGO (IAI-TOGO) est une école inter-états d'enseignement supérieur en Informatique. Il forme en trois (3) ans des ingénieurs des travaux informatiques dans trois filières à savoir : Génie Logiciel (GL), Système & Réseaux (SR) puis Multimédia Technologie Web & Infographie (MTWI). IAI-TOGO propose actuellement le cycle de formation des ingénieurs de travaux informatiques (Licence professionnelle en informatique).

Dans le but de jauger les compétences acquises au cours de nos trois années de formation, de nous permettre d'accroître nos compétences dans les différents domaines de compétence et d'études, et de nous intégrer à la vie professionnelle, l'Institut Africain d'Informatique, Représentation du Togo (IAI-TOGO) prévoit un stage obligatoire de trois (3) mois pour les étudiants de troisième année.

Ce Document est un mémoire de fin de formation en Génie Logiciel, résultat du stage effectué au sein de GESURI Sarl qui a pour l'obtention du Diplôme d'Ingénieur de Travaux Informatiques.

RESUME

Ce mémoire est le résultat du stage de fin de formation effectué au sein de GESURI SARL dans le but d'être attitré Ingénieur des travaux informatiques.

Pendant notre stage, nous avons eu à mettre en place un système capable de gérer les clients, d'entretenir les discussions entre clients et agents, de dresser un historique de discussion, de permettre au client d'effectuer des paiements en lignes et bien d'autres encore. Toutes ces fonctionnalités ont été regroupées dans le thème : « **Gestion des clients, des créances et des discussions instantanées dans un cabinet de conseil en management** ».

Les objectifs de ce système sont multiples mais peuvent être résumé en une seule phrase : **la gestion intégrée du cabinet, depuis le premier contact avec le client jusqu'à la facturation des prestations.**

Ce mémoire, relatara sommairement les étapes qui ont conduit à la mise en place de ce système.

ABSTRACT

This memory memoir is the result of the training period completed at GESURI SARL in order to be appointed as Computer Engineer.

During our internship, we had to set up a system which will be capable to manage of managing clients, maintain maintaining discussions between clients and agents, setup a discussion history, allow customer to make online payments and others. All of these features have been grouped under the theme of **"Managing clients, receivables and instant discussions in a management consulting firm"**.

The objectives of this system are multiple but can be summarized in a single sentence: **the firm's integrated management, from the first contact with the client to the billing of services.**

This brief will summarize all the steps that led to the establishment of this system.

SOMMAIRE

INTRODUCTION GENERALE.....	VIII
LISTE DES PARTICIPANTS	IX
LISTE DES TABLEAUX	X
LISTE DES FIGURES.....	XI
GLOSSAIRE	XIII
PARTIE I : CAHIER DE CHARGE	1
INTRODUCTION	2
I. PRESENTATIONS	3
II. THEME DE STAGE.....	7
III. ETUDE DE L'EXISTANT	9
IV. CRITIQUE DE L'EXISTANT	10
V. PROPOSITION DES SOLUTIONS.....	11
VI. CHOIX DE LA SOLUTION	16
VIII. PLANNING PREVISIONNEL DE REALISATION	17
CONCLUSION	18
PARTIE II : ANALYSE ET CONCEPTION	19
INTRODUCTION	20
I. PRESENTATION DES METHODES D'ANALYSE.....	21
II. PRESENTATION DE L'OUTILS DE MODELISATION	24
III. ETUDE DETAILLE DE LA SOLUTION.....	26
CONCLUSION	56

PARTIE III : MISE EN ŒUVRE ET REALISATION	57
INTRODUCTION	58
I. MISE EN ŒUVRE	59
II. PRÉSENTATION DE L'APPLICATION.....	72
III. PLANNING DE REALISATION DE L'APPLICATION	83
CONCLUSION	84
PARTIE IV : GUIDE D'EXPLOITATION	85
INTRODUCTION	86
I. CONFIGURATION LOGICIELLE ET MATERIELLE	87
II. DEPLOIEMENT ET SUIVI.....	90
CONCLUSION.....	94
PARTIE IV : GUIDE D'UTILISATION.....	95
INTRODUCTION	96
I. PRESENTATION DE L'APPLICATION	97
II. MAINTENANCES.....	102
CONCLUSION	103
CONCLUSION GENERALE	104
TABLE DES MATIERES.....	II

INTRODUCTION GENERALE

L'informatique ne cesse de croître de jours en jours et est devenue un acteur très capital dans le développement d'une entreprise. L'utilisation des outils informatiques apporte un gain considérable de temps et de moyens financiers. Cet outil garantit de même une sécurité, une fiabilité, la réutilisabilité et aussi la disponibilité des données. Vu les nombreux avantages qu'il nous offre, l'outil informatique se retrouve partout et se fait ressentir dans le domaine de la téléphonie.

GESURI SARL de par son évolution fait face à des contraintes. La gestion de la clientèle devenant de plus en plus complexe et vu le nombre sans cesse croissant de celle-ci, il serait mieux d'associer de l'informatique à celle-ci. S'inscrivant dans cet ordre d'idée et désireux de rendre plus facile les échanges avec ses clients le thème « **Gestion des clients, des créances et des discussions instantanées dans un cabinet de conseil en management** » nous a été confié pour notre mémoire de fin de formation.

Ce document divisé en cinq (5) grandes parties à savoir : le cahier de charges, le dossier d'analyse et de conception, le dossier de réalisation et de mise en œuvre, le guide d'exploitation et le guide d'utilisation a pour but de fournir un rapport détaillé des étapes de réalisation de ce projet.

LISTE DES PARTICIPANTS

Tableau 1: liste des participants

NOM ET PRENOMS	ROLES	FONCTIONS
ADUAYOM MESSAN Ekoue Enoc	Stagiaire	Réalisateur
M. AKOUNDA Gbamra	Superviseur	Enseignant à IAI-TOGO
M. DELEGBE Gauss	Maitre de Stage	Consultant à GESURI Sarl

LISTE DES TABLEAUX

<i>Tableau 1: liste des participants.....</i>	<i>ix</i>
<i>Tableau 2: évaluation financière de la solution 1</i>	<i>15</i>
<i>Tableau 3:évaluation financière de la solution 2</i>	<i>15</i>
<i>Tableau 4:récapitulatif des solutions</i>	<i>16</i>
<i>Tableau 5:identification des cas d'utilisations</i>	<i>27</i>
<i>Tableau 6:description textuelle du cas d'utilisations authentification.....</i>	<i>31</i>
<i>Tableau 7:description textuelle du cas d'utilisations discuter.....</i>	<i>33</i>
<i>Tableau 8:description textuelle du cas d'utilisations paiement</i>	<i>35</i>
<i>Tableau 9:description textuelle du cas d'utilisations générer facture.....</i>	<i>37</i>
<i>Tableau 10:description textuelle du cas d'utilisations réclamer paiement</i>	<i>39</i>
<i>Tableau 11:description textuelle du cas d'utilisation réclamer facture</i>	<i>40</i>
<i>Tableau 12: description textuelle du cas d'utilisations ajouter utilisateur.....</i>	<i>42</i>
<i>Tableau 13:description textuelle de cas d'utilisations de suspendre un client</i>	<i>43</i>
<i>Tableau 14: récapitulation de matériels choisis.....</i>	<i>59</i>
<i>Tableau 15:configurations matérielles requise</i>	<i>87</i>
<i>Tableau 16:fiche de renseignement d'erreurs.....</i>	<i>93</i>
<i>Tableau 17:recueil d'erreurs.....</i>	<i>102</i>

LISTE DES FIGURES

Figure 1: Organigramme GESURI	6
Figure 2: Plan de localisation GESURI Sarl.....	6
Figure 3:interface Neogia.....	11
Figure 4:Cegid interface	11
Figure 5: application de discussion.....	13
Figure 6:planning prévisionnel	17
Figure 7: logo UML	21
Figure 8: Processus de développement en Y	23
Figure 9: logo Power Designer	24
Figure 10: actions des clients sur le système	28
Figure 11: actions des agents sur le système	29
Figure 12: diagramme de cas d'utilisations général	30
Figure 13: diagramme de séquences de paiement.....	45
Figure 14: diagramme de séquences de discussion.....	46
Figure 15: diagramme de séquences d'ajout d'utilisateur	47
Figure 16: diagramme de séquences d'authentification	48
Figure 17: diagramme d'activités de discussion.....	49
Figure 18: diagramme d'activités de paiement.....	50
Figure 19: diagramme d'activités de gestion	51
Figure 20: diagramme d'activités d'authentification	52
Figure 21:gestion des messages.....	53
Figure 22:diagramme de classe (administration)	53
Figure 23:gestion de facture	54
Figure 24: diagramme de classe.....	55
Figure 25:logo LARAVEL	59
Figure 26:logo GIT.....	61
Figure 27:logo BITBUCKET.....	61
Figure 29:logo PHP Storm	62
Figure 31:logo MySQL	63
Figure 32: logo Socket	63
Figure 33:logo Node.js	65
Figure 34:Mustache	66

Figure 35:logo material design	66
Figure 36:logo JQuery	67
Figure 37: architecture de l'application.....	68
Figure 38:architecture MVC	70
Figure 39: maquette de l'application	72
Figure 41: structure de l'application	73
Figure 42:formulaire d'ajout d'agent	77
Figure 43:discussion instantanée	80
Figure 44:statistique aperçu global.....	81
Figure 45: aperçu individuel	81
Figure 46:aperçu en bâton	82
Figure 47:planning de réalisation.....	83
Figure 48: installation Redis	89
Figure 49:Fiche de renseignement d'erreur.....	92
Figure 50:page d'accueil	97
Figure 51:page de connexion	98
Figure 52:menu de l'administrateur.....	98
Figure 53:menu d'un agent	99
Figure 54:menu modal	100
Figure 55:info client.....	101
Figure 56:info agents	101

GLOSSAIRE

DBE : Développement Back end

ERP : Enterprise Resource Planning

PARTIE I

CAHIER DE CHARGE

Du 19 Juillet 2017 Au 31 Juillet 2017

INTRODUCTION

Le cahier de charge est un document qui contient les détails bien décrits des besoins de l'utilisateur. C'est également le recueil de l'étude de faisabilité en allant de l'existant à l'étude des solutions disponibles pour en tirer la meilleure. Ce document contient également le planning d'élaboration prévisionnel du projet.

Il permet aux différentes parties prenantes de s'accorder sur les spécificités des objectifs et résultats attendus du projet. Il fera, l'objet de référence lors de l'évaluation du produit fini. Dans ce premier document, nous nous axerons sur la présentation du contexte de stage, le thème de stage, l'étude de l'existant, la critique de l'existant, les propositions de solutions, le choix de solution et le planning prévisionnel de réalisation.

I. PRESENTATIONS

1. Présentation de l'IAI-TOGO

L'Institut Africain d'Informatique, représentation du TOGO (IAI-TOGO) est ouvert le 22 octobre 2002. IAI-TOGO est une école inter-états d'enseignement supérieur en informatique. Il est membre du réseau IAI créé le 29 janvier 1971 à Fort Lamy (actuel Ndjamena) en république de TCHAD. L'accord d'établissement entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006.

IAI-TOGO forme en trois (3) ans des ingénieurs des travaux informatiques dans trois filières : Génie Logiciel (GL), Système & Réseaux (SR) et Multimédia Technologie Web & Infographie (MTWI). IAI-TOGO propose actuellement le cycle de formation des ingénieurs de travaux informatiques (Licence professionnelle en informatique) et de Master. Au terme des trois années de formation, les diplômés peuvent poursuivre leurs études supérieures à l'IAI-TOGO ou dans le pays de leur choix.

Il propose aussi une formation dénommée formation Cisco. Les cours Cisco sont découpés en quatre (4) modules CCNA1, CCNA2, CCNA3 et CCNA4, tous accessibles via Internet.

En ce qui concerne l'organisation structurelle de IAI-TOGO, nous avons :

- ☞ M. AGBETI A. B. Kodjo, Représentant Résident de l'IAI-TOGO. Il est l'intermédiaire entre l'IAI-Gabon et celui du Togo ;
- ☞ M. AMEYIKPO Kossi N. Directeur des Affaires Académiques et scolaires (DAAS) est chargé des problèmes liés aux études entre autres le respect du règlement intérieur ;
- ☞ Les secrétaires qui servent d'intermédiaire entre l'administration et les étudiants.

2. Présentation de GESURI Sarl

2.1 Statut

Le cabinet Gestion Suppléance Recherches Intermédiation (GESURI Sarl) est un cabinet privé de management. Fondé en avril 2000 par une équipe de consultants expérimentés et engagée à promouvoir l'excellence dans la gestion des organisations, GESURI s'est spécialisé dans l'appui-accompagnement des entreprises, des projets et organisations de développement, par la mise à leur disposition des services d'études, d'appui conseil et de formation.

2.2 Mission

La principale mission de GESURI Sarl est de mettre son expertise au service de la culture de l'excellence dans le management des organisations.

Les prestations qu'offre GESURI s'adressent prioritairement :

- ☞ Aux PME/PMI,
- ☞ Aux Organisations de Développement,
- ☞ Et à tous promoteurs d'affaires.

2.3 Activités

Les prestations de GESURI Sarl sont proposées en 03 paquets professionnels :

- ❖ Conseils en Organisation et Ressources Humaines (CORHU)
 - Recrutement
 - Manuel de procédures
 - Réglementation du travail
 - Audit organisationnel
 - Bilan de compétence et plan de formation en entreprise
- ❖ Suivi Comptable et Fiscal (SUCOF)

- Assistance mensuelle aux Entreprises
- Revue annuelle de la comptabilité et des déclarations fiscales
- Revue ponctuelle de la comptabilité et des déclarations fiscales
- ❖ Études socio-économiques et Renforcement des Capacités (ERCA)
 - Faisabilité et montage de dossier financier
 - Évaluation de projet
 - Plan d'affaires

Outre ces prestations suivant les besoins des partenaires et clients, GESURI propose un panel de programmes annuels de séminaire-atelier dont :

- ❖ L'art d'entreprendre et le rôle du conseiller d'entreprise
- ❖ Les choix stratégiques pour la pérennité d'une organisation

Chacun de ces paquets de prestations fait l'objet d'un département dans l'organisation du travail au sein du cabinet GESURI. La conduite de ces différentes missions suit à chaque fois et pour chaque cas des procédures spécifiques rigoureuses mais flexibles.

2.4 Quelques réalisations

GESURI dans ses activités, se charge du suivi fiscal de certains clients tels que les entreprises commerciales, les entreprises industrielles, les restaurants, les églises, les boutiques, les cliniques, les bars et les salles de gym etc... et permet à certains contribuables d'être en règle. Quelques-uns de ses clients sont :

- CCT Batimât
- MBC alimentation
- Privilège
- Coco Beach
- BCBG

II. THEME DE STAGE

1. Présentation du sujet

Durant la période d'intégration au sein de l'entreprise, nous avons eu à étudier comment se faisait les échanges entre les clients et la société.

Dans l'optique d'optimiser les prestations envers ses clients, GESURI Sarl a jugé bon de nous confier comme thème de stage : la « **Gestion des clients, des créances et des discussions instantanées dans un cabinet de conseil en management** ».

Notre travail consistera ainsi à identifier les faiblesses et les points forts des opérations actuelles entre les clients et la société ainsi, proposer un système informatique qui pourra tirer le plus de ces forces tout en réduisant au maximum les faiblesses que celle-ci présente.

2. La problématique du sujet

Les problèmes de notre centre d'accueil tournent autour de ces quelques questions :

- ❖ Comment gérer efficacement ses clients ? (Comment avoir des informations solides, disponibles et accessibles à tout instant sur chaque client ?)
- ❖ Comment conseiller instantanément les clients ? (Comment répondre plus facilement, plus rapidement et instantanément aux besoins des clients ?)
- ❖ Comment suivre et permettre aux clients de suivre l'état des créances, les paiements ?

Au vu de tout cela et dans le souci de répondre à toutes ces questions, GESURI nous a confié la mission d'étudier ses difficultés et de proposer une solution informatisée adaptée à son entreprise.

3. Intérêt du sujet

a. Objectifs

○ Objectif général

L'objectif principal de notre système est de permettre une gestion intégrée du cabinet, depuis le premier contact avec le client jusqu'à la facturation des prestations.

○ Objectifs spécifiques

Les objectifs secondaires de notre application sont les suivantes :

- Rendre disponible les données sur chaque client de l'entreprise,
- Permettre la discussion instantanée entre le client et un conseiller de l'entreprise,
- Facturer les prestations,
- Permettre aux clients d'effectuer des paiements par carte bancaire ou carte de crédit,
- Notification par sms de l'agent.

b. Résultats attendus

Après la conception de notre application, l'utilisateur selon le niveau d'habilitation, doit pouvoir :

- Consulter les factures uploadées par les clients
- Avoir les historiques de connexion de tous les clients
- Dresser un historique des discussions
- Notifier les agents hors ligne par sms
- Consulter la plupart des informations en locale
- Envoyer des notifications de rappel au client
- Suspendre un client

III. ETUDE DE L'EXISTANT

Le cadre qui nous accueille pour notre stage ne dispose pas encore de moyen informatique pour gérer sa clientèle, tout se fait de façon manuelle jusqu'alors.

- ❖ En ce qui concerne les discussions entre GESURI et ses clients, cela se fait le plus souvent en présentiel. Le client appelle le cabinet, soumet ses inquiétudes et ce dernier les prend en compte.
- ❖ Pour enregistrer un client, GESURI :
 - Prend contact avec celui-ci ;
 - S'entend sur les conditions de partenariat ;
 - Signe un contrat avec celui-ci.
- ❖ GESURI, pour faire le suivi comptable et fiscal de ses clients et leur permettre d'être en règle conformément aux dispositions en la matière, se rend chaque semaine chez ces derniers pour récupérer toutes les pièces comptables (exemple de pièces comptables : les factures d'achat et de vente, etc.) Exceptionnellement pour certains clients, le traitement des pièces se fait sur place par le conseiller.
- ❖ La transmission des documents comptables se fait de main à main pour chaque client.

IV. CRITIQUE DE L'EXISTANT

Les pratiques actuelles de GESURI font face le plus souvent à des contraintes, si ce n'est des contraintes temporelles ou géographiques, ce sont celles de disponibilité de l'agent lui-même ou de coûts de déplacement pour collecter les factures.

De quoi souffrent ces pratiques ?

- Les données sur les clients ne sont pas convenablement archivées,
- Il n'y a pas de suivi des créances du client,
- En cas d'urgence, le client n'a la possibilité de contacter le cabinet que par appel téléphonique, si le conseiller est disponible en ce moment,
- Tous ces déplacements pour récupérer les factures chez les différents clients ne sont pas vraiment intéressants,
- L'accumulation des données en papiers peuvent ralentir le fonctionnement de l'entreprise,
- L'insécurité des données, car tout le monde peut y avoir accès,
- Le fait de transmettre tous les documents de façon manuelle ne fait pas du tout gagner en temps et en ressources financières à GESURI ; de même que pour les transactions monétaires qui ne sont pas sécurisées.

V. PROPOSITION DES SOLUTIONS

1. Évaluations techniques

Solution 1 : Acheter un logiciel de gestion intégré qui pourra à la fois gérer tout le cabinet et tous les clients et y combiner une application de discussion instantanée.

- **Progiciel intégré :**
 - **Neogia (libre)**

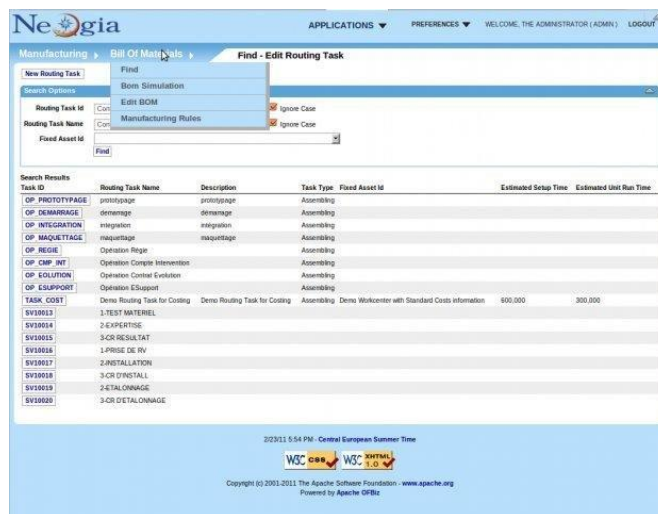


Figure 3: interface Neogia

- **Cegid**

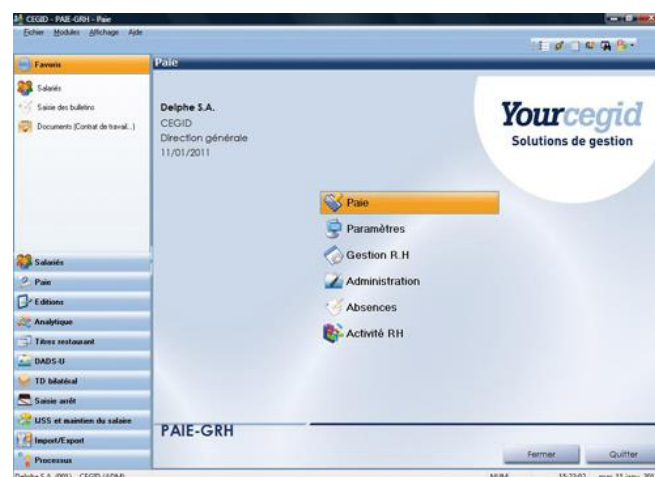


Figure 4: Cegid interface

Cette solution semble être la plus efficace au vue de ses avantages :

- L'optimisation des processus de gestion (flux économiques et financiers),
- La cohérence et l'homogénéité des informations (un seul fichier « articles », un seul fichier « clients », etc.) qui permettent de respecter des normes,
- L'intégrité et l'unicité du système d'information,
- Le partage du même système d'information facilitant la communication interne ou la mobilité,
- La globalisation de la formation (même logique, même ergonomie),
- Une aide à la productivité,
- Un contrôle centralisé de l'entreprise,
- Une aide à la prise rapide de décision.

Mais ses inconvénients sont aussi multiples :

- Elle demande de l'organisation et de la rigueur,
- Sa mise en œuvre reste complexe,
- La bonne connaissance et la remise en cause des processus de l'entreprise,
- Le coût : de 5 000 à 12 000 € par utilisateur (matériel + licence + intégration + formation + maintenance),
- Des difficultés d'appropriation par le personnel de l'entreprise liées aux changements,
- Tout ERP a au moins un talon d'Achille : les modules sont de valeurs inégales au vu de ses inconvénients.

- **Application de discussion : Microsoft Exchange, SharePoint, Skype Entreprise.**



Figure 5: application de discussion

- Les points forts : Microsoft est clairement reconnu comme un éditeur de référence dans le monde des solutions collaboratives dédiées aux entreprises. Composées du service de messagerie instantanée Skype Entreprise (anciennement Lync), de la plateforme SharePoint dédiée aux Intranets, Extranets, sites web ou de la messagerie email Exchange, Microsoft dispose d'une véritable expérience lorsqu'il s'agit de fournir des outils collaboratifs s'intégrant idéalement dans l'environnement informatique d'une PME et d'un grand groupe. D'ailleurs de nombreuses entreprises ont fait le choix d'intégrer ces solutions collaboratives à la base de leurs propres offres. Ainsi, des sociétés fournissent aujourd'hui des services additionnels et complémentaires à ce type de solution : prestations de développement personnalisé, support technique avec des temps de disponibilité modulable jusqu'à 24h/7j, accompagnement à l'adoption des nouveaux usages collaboratifs ou formations.
- Les points faibles : L'unification des différentes solutions collaboratives, selon qu'elles soient hébergées en mode cloud ou « on-premise », peut s'avérer complexe et nécessiter un accompagnement spécifique pour y

parvenir. D'où la volonté de Microsoft, en partie, d'accentuer fortement sa communication et d'orienter les utilisateurs vers Office 365 et sa suite collaborative. Mis à part un prix parfois plus élevé que d'autres solutions collaboratives concurrentes, pouvant aussi s'expliquer chez certains fournisseurs par des services complémentaires apportés aux solutions de base, les outils collaboratifs Microsoft bénéficient toujours d'une forte renommée et sont largement adoptés auprès de nombreuses PME, grands groupes et établissements publics.

Solution 2 : Développer une application de gestion intégrant la discussion instantanée, à moindre coût et facilement paramétrable.

Avantages :

- Application propre aux besoins de l'entreprise,
- Optimisé pour traiter les demandes de l'entreprise,
- Maintenance facile et à prix réduit,
- Déploiement et acquisition à prix réduit,
- Combinaison de toutes les fonctionnalités dans une seule application,
- Possibilité de faire des mises à jour propres à l'entreprise,
- Ouvert au client.

Inconvénients :

- Demande beaucoup de temps,
- Implique d'engager un développeur d'application.

2. Évaluations financières

Solution 1

Tableau 2: évaluation financière de la solution 1

Libellé	Coût pour plus de 100 utilisateurs	
	Libre	Payant
Licence	0	3 324 125 FCFA
Intégration du progiciel	1 424 625 FCFA	
Infrastructure	4 748 750 FCFA	
Maintenance	2 374 375 FCFA	
Total	8 545 750 FCFA	11 871 875 FCFA

Solution 2

Tableau 3:évaluation financière de la solution 2

Matériels	Prix unitaires	Quantité	Prix total
Carte SIM (pour test)	1 000 FCFA	3	3 000 FCFA
Modem professionnel	60 000 FCFA	1	60 000 FCFA
Ordinateur Ultrabook	786 492,44 FCFA	1	786 492,44 FCFA
Téléphone Android	286 389,19	1	286 389,19 FCFA
Serveur dédié Processeur Intel Xeon Core i7 3.5GHz, RAM 32Go, Disque 512Go SSD + 4*500Go HDD	52 000 FCFA/mois	-	624 000 FCFA /an
TOTAL			1 759 881,63 FCFA

VI. CHOIX DE LA SOLUTION

Tableau 4:récapitulatif des solutions

	AVANTAGES	INCONVENIENTS
SOLUTION 1	<ul style="list-style-type: none"> - Gain de temps, - Aide à la prise de décision, - Aide à la productivité, - contrôle centralisé de la société. 	<ul style="list-style-type: none"> - Coût élevé d'acquisition et de maintenance, - Combinaison de deux applications, - Coût de formation, - Mise en œuvre complexe, - modules inutiles disponibles.
SOLUTION 2	<ul style="list-style-type: none"> - Application propre aux besoins de l'entreprise, - Optimisée pour traiter les demandes de l'entreprise, - Maintenance facile et à prix réduit, - Déploiement et acquisition à prix réduit, - Combinaison de toutes les fonctionnalités dans une seule application, - possibilité de faire des mise à jour propres à l'entreprise. 	<p>Demande beaucoup plus de temps et de main d'œuvre</p>

Vues les demandes du cabinet et suite à la confrontation des deux solutions, nous retenons la solution 2 qui est, de développer une application.

VIII. PLANNING PREVISIONNEL DE REALISATION

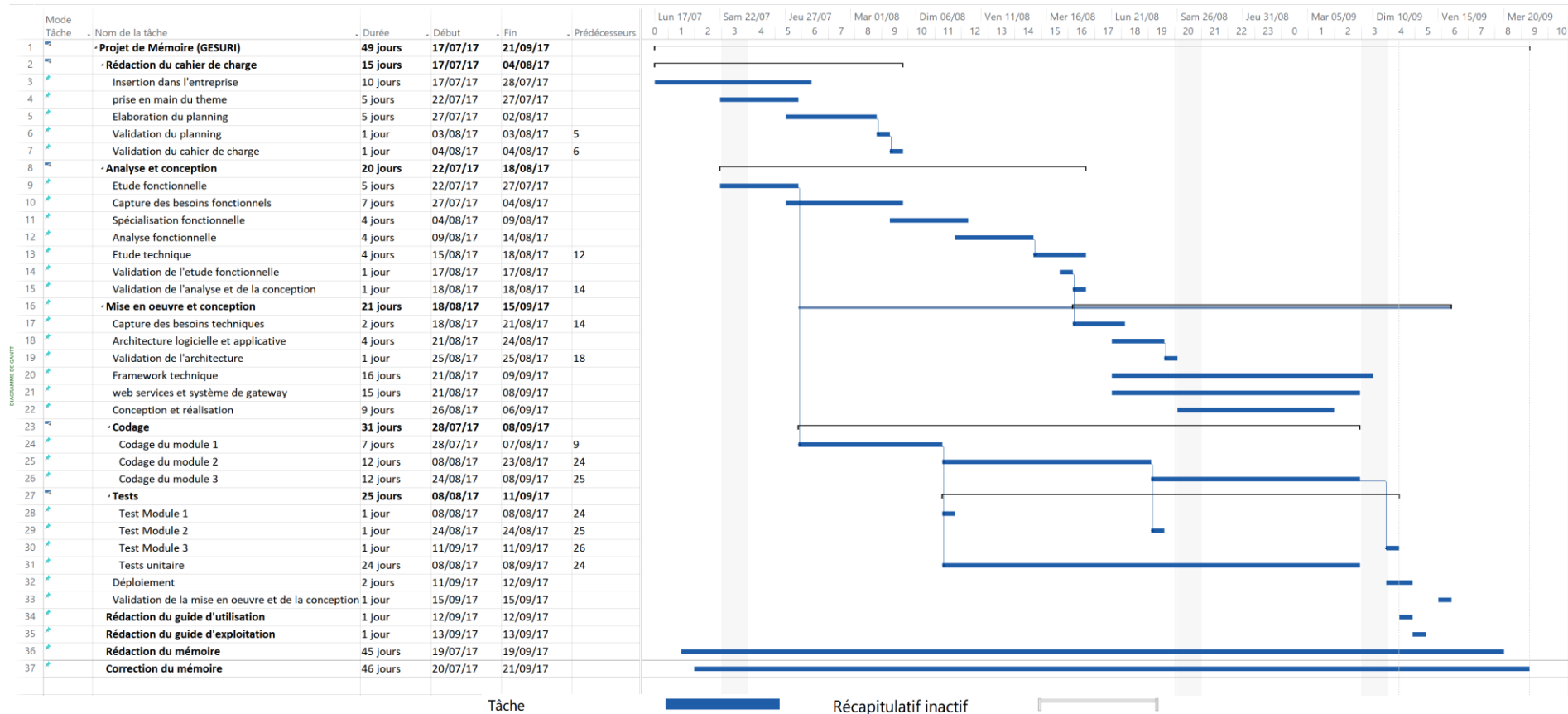


Figure 6:planning prévisionnel

CONCLUSION

Le cahier de charge présente de façon globale le système à mettre en place. Il nous fournit les informations nécessaires pour le développement et la conception du futur système.

En effet dans cette première partie, il a été question de présenter GESURI Sarl, notre cadre de stage. Nous avons ensuite présenté les grandes lignes et fonctionnalités de nos projets en le confrontant aux pratiques existantes et à d'autres solutions éventuelles. Nous allons aborder maintenant la seconde partie qui est intitulée : analyse et conception.

PARTIE II

ANALYSE ET CONCEPTION

Du 01 Août 2017 Au 28 Août 2017

INTRODUCTION

L'analyse et la conception consiste à étudier le projet pour poser les bases du thème.

Dans cette partie, nous allons présenter les méthodes d'analyse et l'outil d'analyse choisi. Nous allons ensuite présenter le modèle d'analyse et les diagrammes qui nous permettront de mettre en place efficacement le système que nous a été confié.

I. PRESENTATION DES METHODES D'ANALYSE

1. Langage UML



Figure 7: logo UML

Le **langage de modélisation unifié**, de l'anglais Unified Modeling Language (**UML**), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML dispose de différents diagrammes pour modéliser le système. Les diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Il en existe 14 (depuis UML 2.3 contre 9 pour UML 1.3).

- ❖ Les diagrammes structurels ou statiques (07) :
 - Diagramme de classes (class diagram) ;
 - Diagramme d'objets (object diagram) ;
 - Diagramme de composants (component diagram) ;
 - Diagramme de déploiement (deployment diagram) ;
 - Diagramme des paquets (package diagram) ;

- Diagramme de structure composite (composite structure diagram) ;
- Diagramme de profils (profile diagram).
- ❖ Les diagrammes comportementaux (03) :
 - Diagramme des cas d'utilisation (use-case diagram) ;
 - Diagramme états-transitions (state machine diagram) ;
 - Diagramme d'activité (activity diagram).
- ❖ Les diagrammes d'interaction ou dynamiques (04) :
 - Diagramme de séquence (sequence diagram) ;
 - Diagramme de communication (communication diagram) ;
 - Diagramme global d'interaction (interaction overview diagram) ;
 - Diagramme de temps (timing diagram).

2. Processus 2TUP

2TUP (2 tracks unified process), est un processus développement logiciel qui met en œuvre la méthode du processus Unifié. Le **2TUP** propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de trois phases essentielles :

- Une branche technique ;
- Une branche fonctionnelle ;
- Une phase de réalisation.

La branche fonctionnelle capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.

La branche technique capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.

La phase de réalisation consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

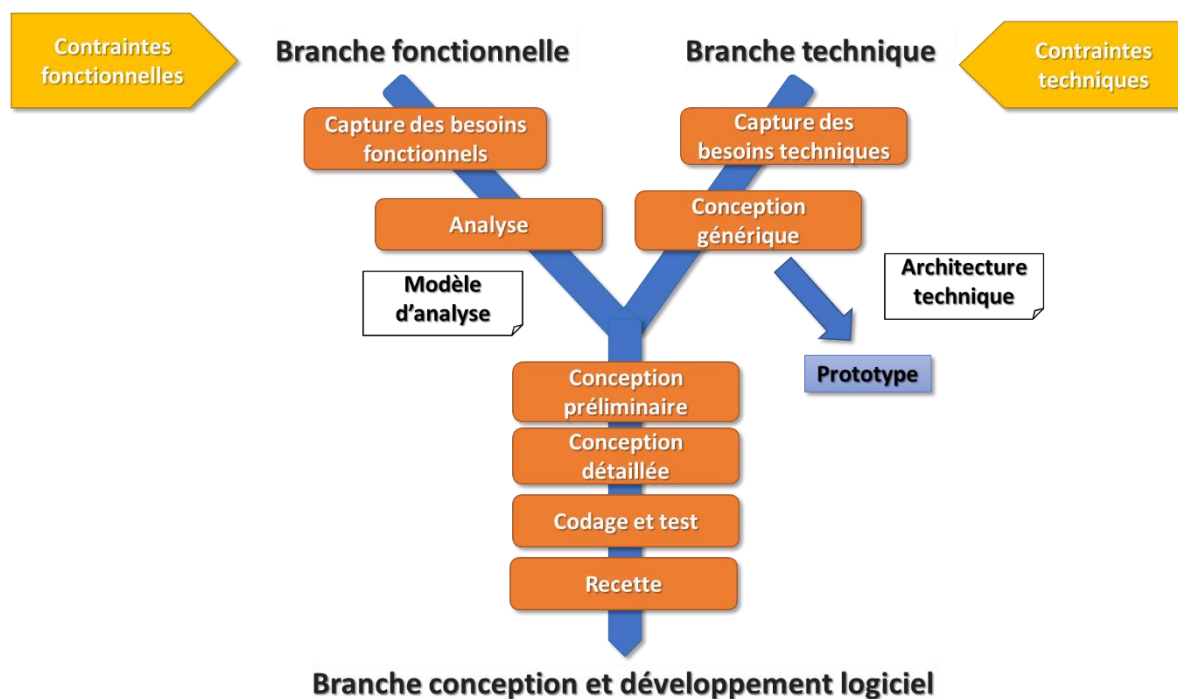


Figure 8: Processus de développement en Y

II. PRESENTATION DE L'OUTILS DE MODELISATION

L'outil retenu pour la modélisation est Sybase POWERAMC Version 15.1.

1. Qu'est-ce que Power Designer



Figure 9: logo Power Designer

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC Designer, racheté par Power soft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque Power Designer.

Il permet d'effectuer les tâches suivantes :

- ❖ Modélisation intégrée via l'utilisation de méthodologies et de notation standard :
 - Données (E/R, Merise) ;
 - Métiers (BPMN, BPEL, ebXML) ;
 - Application (UML) ;
- ❖ Génération automatique de code via des Template personnalisables :
 - SQL (avec plus de 50 SGBD) ;
 - Java ;
 - .net ;
- ❖ Fonctionnalités de reverse engineering pour documenter et mettre à jour des systèmes existants ;

- ❖ Une solution de référentiel d'entreprise avec des fonctionnalités de sécurité et de gestion des versions très complètes pour permettre un développement multiutilisateur ;
- ❖ Fonctionnalités de génération et de gestion de rapports automatisés et personnalisables ;
- ❖ Un environnement extensible, qui vous permet d'ajouter des règles, des commandes, des concepts et des attributs à vos méthodologies de modélisation et de codage.

2. Modélisation avec Power Designer

Power Designer fournit un jeu unique d'outils de modélisation professionnels qui associent les techniques et notations standards de la modélisation de processus métiers, de la modélisation des données et de la modélisation d'applications UML et d'autres fonctionnalités complexes afin d'aider à analyser, à concevoir, à construire et à maintenir des applications, en utilisant les techniques les plus élaborées d'ingénierie logicielle. La solution de modélisation Power Designer permet d'intégrer étroitement la conception et la maintenance des couches de données centrales de l'application et exigences de projet, processus métiers, code orienté objet, vocabulaires XML et informations de réplication de base de données. En mettant à disposition une série complète de modèles à tous les niveaux d'abstraction, Power Designer permet d'élargir la portée du processus de conception itérative à tous les aspects de l'architecture du système, de la conception au déploiement, et même au-delà.

III. ETUDE DETAILLE DE LA SOLUTION

1. Diagrammes de cas d'utilisation

Un diagramme de cas d'utilisation définit toutes les fonctionnalités du système à concevoir. Avant de présenter ces fonctionnalités, nous allons d'abord identifier les différents acteurs (entité interagissant avec le système).

1.1 Identification des acteurs

Nous distinguons trois grandes catégories d'acteurs :

- ❖ Les agents (qui modifient l'état du système ou qui le consultent) :
 - L'administrateur (administre le système)
 - Les autres employés de l'entreprise
- ❖ Les Clients (Ils nourrissent le système et apportent les éléments nécessaires pour que le système fonctionne correctement) :
 - Le directeur Général
 - Le chef comptable
- ❖ Les promoteurs (de futurs clients)

1.2 Identification des cas d'utilisation

Les cas d'utilisations sont les suivants :

Tableau 5:identification des cas d'utilisations

Acteur(s)	Cas d'utilisation
AGENTS/ ADMINISTRATEURS	Enregistrer un client
	Modifier un client
	Suspendre un client
	Consulter les créances d'un client
	Consulter les factures d'un client
	Consulter une histoire
	Suivre l'évolution d'un client
	Editer une facture
	Imprimer une facture
	Réclamer paiement
CLIENTS	Consulter ses factures
	Réclamer facture
	effectuer paiement
	Consulter historique
	Consulter créance
	Suivre son évolution
	Demander conseil
CLIENTS, PROMOTEURS & AGENTS	Envoyer message
	Consulter message

1.3 Les diagrammes de cas d'utilisation

- Actions des clients sur le système

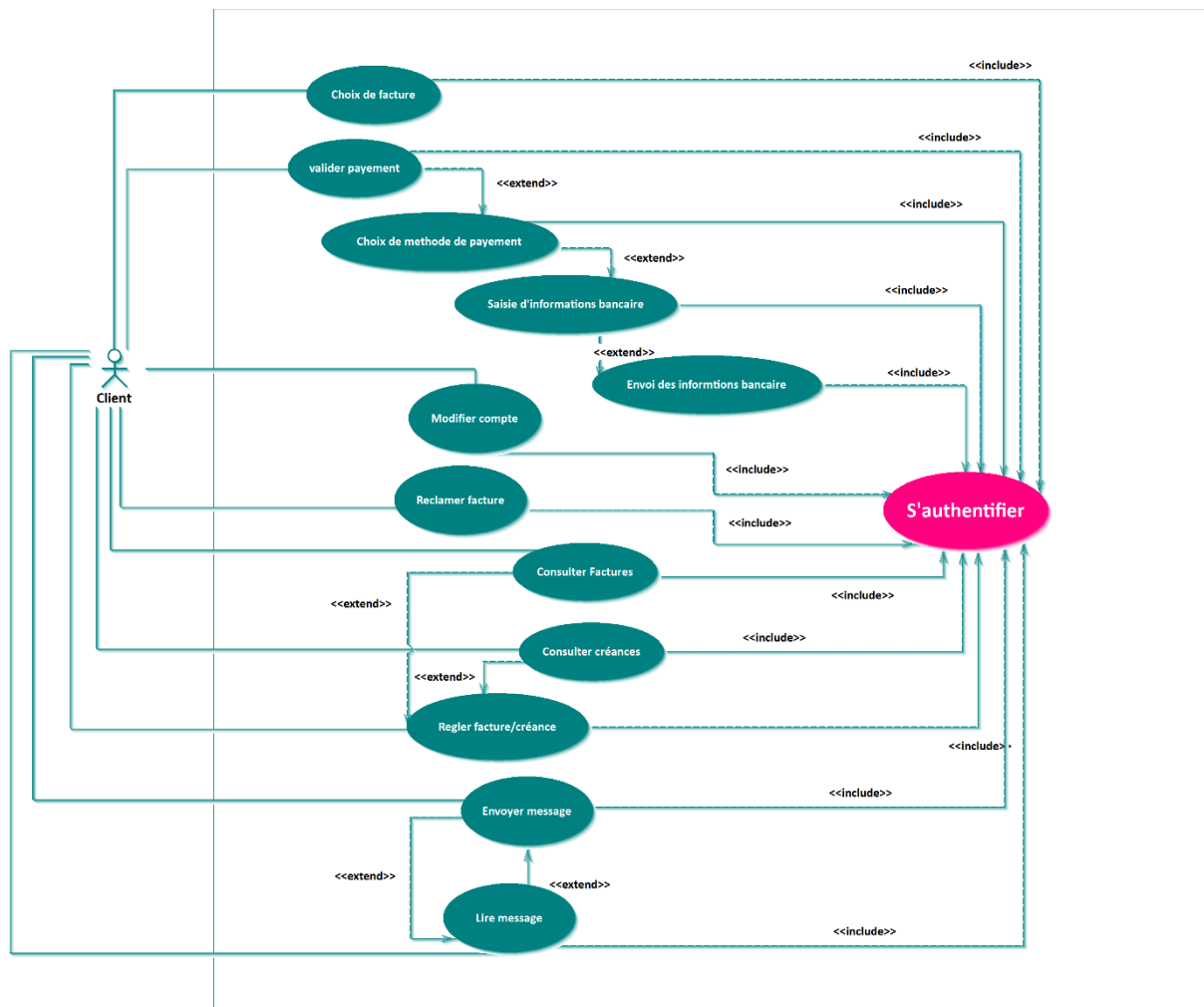


Figure 10: actions des clients sur le système

- Actions des agents sur le système

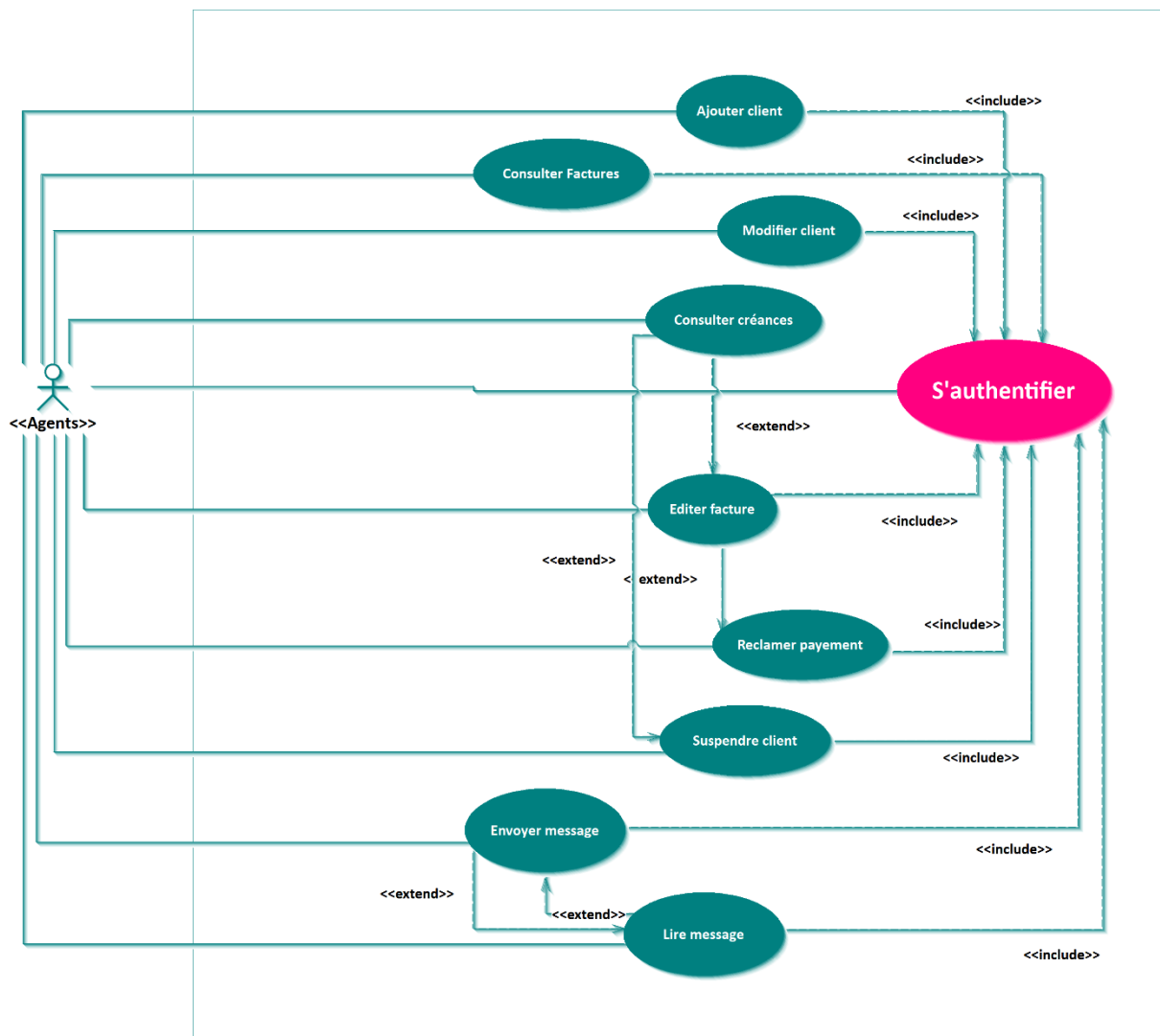


Figure 11: actions des agents sur le système

- Diagramme de cas d'utilisation général

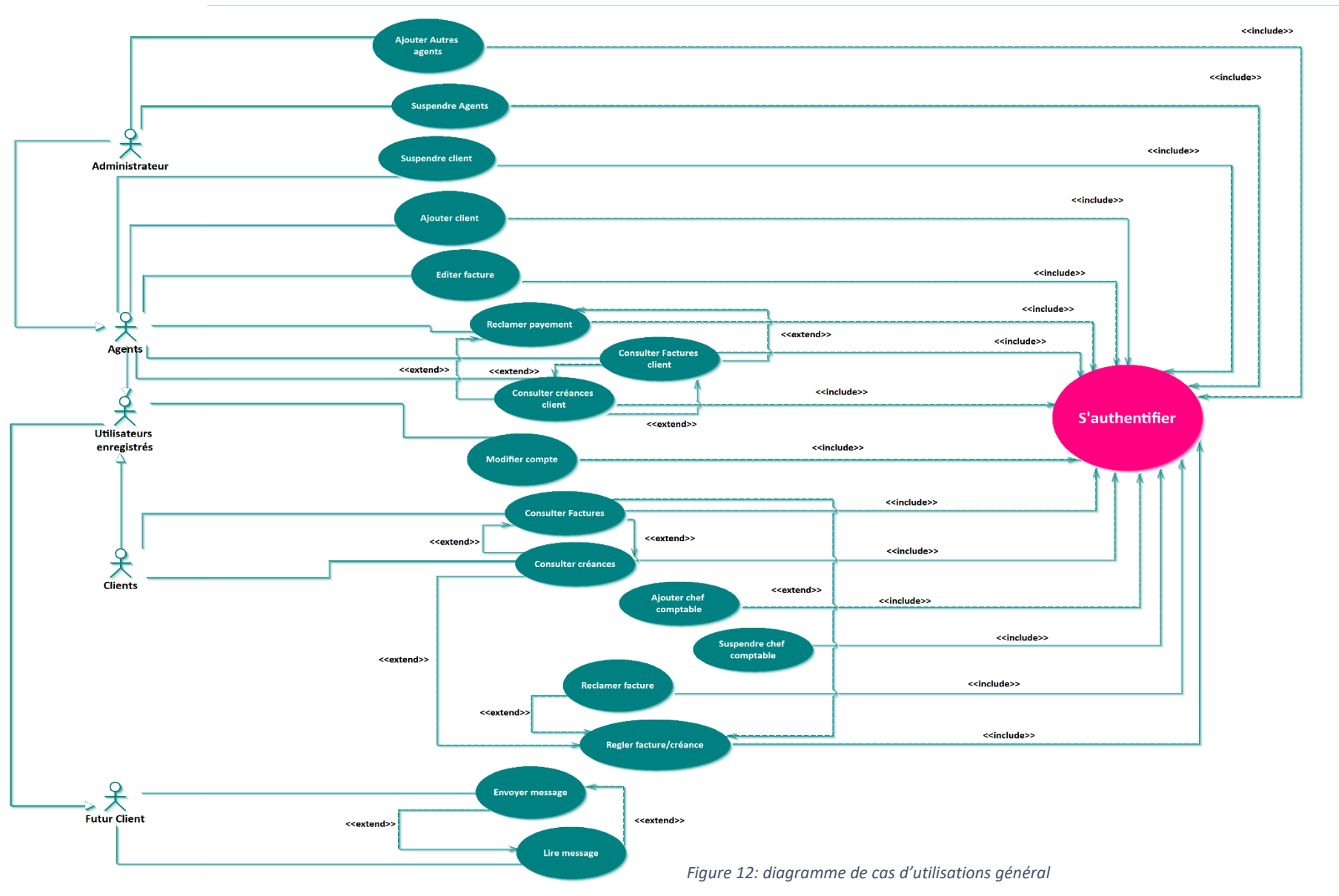


Figure 12: diagramme de cas d'utilisations général

1.4 Description textuelle des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'interactions entre le système et ses acteurs. Ainsi, la description de la dynamique du cas d'utilisation consiste à recenser toutes les interactions de façon textuelle. Le cas d'utilisation doit par ailleurs avoir un début et une fin clairement identifiés. Il doit préciser quand ont lieu les interactions et quels sont les messages échangés. Il faut également préciser les variantes possibles, telles que les différents cas nominaux, les cas alternatifs, les cas d'erreurs ou d'exceptions, tout en essayant d'ordonner séquentiellement les descriptions, afin d'améliorer leur lisibilité.

- L'authentification

Tableau 6: description textuelle du cas d'utilisations authentification

Sommaire d'identification
<p><u>Titre</u> : S'authentifier</p> <p><u>Acteurs</u> : Agents, Clients</p> <p><u>Résumé</u> : Permet à l'utilisateur de se connecter à la plateforme</p> <p><u>Responsable</u> : ADUAYOM MESSAN E. Enoc</p> <p>Version : 1.0</p> <p><u>Date</u> : 04/08/2017</p>
Description des scénarii
❖ Pré condition
<p>✓ Le système est fonctionnel ;</p> <p>✓ La page de connexion est accessible.</p>

❖ Scénario nominal

1. L'utilisateur renseigne les informations de connexion ;
2. L'utilisateur valide les informations renseignées ;
3. Le système vérifie la conformité des informations renseignées ;
4. Le système affiche la page d'accueil selon les droits qui sont assignés à l'utilisateur.

❖ Scénario d'exceptions

- E1 : L'utilisateur annule la connexion
 - ✓ Ce scénario débute au point 1 du scénario nominal lorsque l'utilisateur annule la connexion ;
 - ✓ Le scénario nominal est interrompu.
- E2 : L'utilisateur saisie les informations de connexion erronées à (3) reprises
 - ✓ Ce scénario commence au point 3 du scénario nominal lorsque le système remarque que les informations saisies sont incorrectes ;
 - ✓ L'utilisateur est informé de l'épuisement des essais
 - ✓ Le système quitte la page de connexion ;
 - ✓ Le système verrouille momentanément l'accès à cet utilisateur (compte inactif).

❖ Scénario alternatif

- A1 : Les informations sont erronées moins de 3 fois

- ✓ Il débute à l'étape 3 du scénario nominal lorsque l'utilisateur saisi des informations erronés ;
- ✓ L'utilisateur reçoit un message qui lui affiche le nombre d'essais restants.

❖ Scénario d'exceptions

- E3 : Les informations sont erronées plus de 3 fois
- ✓ Ce scénario débute à l'étape 3 lorsque les informations erronées dépassent 3 essais ;
- ✓ Le système informe l'utilisateur que le nombre d'essais maximum est atteint et lui demande de contacter l'administrateur pour plus d'informations ;
- ✓ Le compte est désactivé.

❖ Post conditions

- ✓ L'utilisateur est connecté ;
- ✓ Le journal est mis à jour ;
- ✓ Le système est toujours fonctionnel.

- Discuter

Tableau 7: description textuelle du cas d'utilisations discuter

Sommaire d'identification

Titre : Discuter

Acteurs : Agents, Clients

Résumé : Permet aux utilisateurs d'échanger entre eux

Responsable : ADUAYOM MESSAN E. Enoc

Version : 1.0

Date : 04/08/2017

Description des scénarii

❖ Pré condition

- ✓ Le système est fonctionnel ;
- ✓ L'utilisateur est connecté.

❖ Scénario nominal

1. Le client écrit à un agent de l'entreprise ;
2. Le système historique le message ;
3. Le système vérifie l'état de connexion du récepteur ;
4. L'agent renvoi une réponse;
5. Le système historique le message
6. le système vérifie l'état de connexion du récepteur.

❖ Scénario d'exceptions

- E1 : L'agent est offline
- ✓ Ce scénario débute au point 3 du scénario nominal;
- ✓ Le système envoi un sms de notification à l'agent;
- ✓ Le scénario nominal reprend.
- E2 : Le client est offline
- ✓ Ce scénario commence au point 6 du scénario nominal ;

✓ Le système laisse une notification au client;

✓ Le scénario nominal reprend.

❖ Scénario alternatif

- A1 : Le client répond au message de l'agent

✓ Il débute après l'étape 6 du scénario nominal lorsque le client reçoit une réponse de l'agent;

✓ Le scénario nominal reprend depuis le début.

- A2 : Le client clôt la discussion

✓ Il débute après l'étape 6 du scénario nominal lorsque le client reçoit une réponse de l'agent;

✓ Le scénario nominal est interrompu.

❖ Post conditions

✓ L'utilisateur reçoit le message ;

✓ Le journal est mis à jour.

- Payer

Tableau 8: description textuelle du cas d'utilisations paiement

Sommaire d'identification

Titre : Payer

Acteurs : Clients

Résumé : Payer au client de payer ses factures/ses créances

Responsable : ADUAYOM MESSAN E. Enoc

Version : 1.0

Date : 04/08/2017

Description des scénarii

❖ Pré condition

- ✓ Le système est fonctionnel ;
- ✓ L'utilisateur est connecté.

❖ Scénario nominal

1. Le client consulte ses factures/ses créances ;
2. Le client choisit un/des factures à payer ;
3. Le client fait le choix du mode de paiement ;
4. Le client saisie les informations bancaire;
5. Le système vérifie l'état du compte du client;
6. le système valide et historie le payement.

❖ Scénario d'exceptions

- E1 : Les informations bancaires saisie sont incorrectes
 - ✓ Ce scénario débute au point 4 du scénario nominal;
 - ✓ Le système avertit le client que les informations saisies sont incorrectes;
 - ✓ Le scénario nominal est interrompu.
- E2 : L'état du compte ne permet pas au client d'effectuer le payement
 - ✓ Ce scénario commence au point 6 du scénario nominal ;

✓ Le système renvoie le client vers le point 3 du scénario nominal;
✓ Le scénario nominal continu.
❖ Scénario alternatif
<ul style="list-style-type: none"> • A1 : Le client annule le paiement
✓ Il débute après l'étape 5 du scénario nominal;
✓ Le scénario nominal est interrompu.
❖ Post conditions
✓ Le paiement est effectué ;
✓ Le journal est mis à jour.

- Générer facture

Tableau 9: description textuelle du cas d'utilisations générer facture

Sommaire d'identification
<p><u>Titre</u> : Générer facture</p> <p>Acteurs : Agents</p> <p><u>Résumé</u> : Permet de demander au système de faire la somme et les détails de certaines prestations</p> <p><u>Responsable</u> : ADUAYOM MESSAN E. Enoc</p> <p>Version : 1.0</p> <p><u>Date</u> : 04/08/2017</p>
Description des scénarii

❖ Pré condition
<ul style="list-style-type: none"> ✓ Le système est fonctionnel ; ✓ L'utilisateur est connecté.
❖ Scénario nominal
<ol style="list-style-type: none"> 1. L'agent consulte Les prestations effectuées ; 2. Il choisit les prestations dont il veut générer la facture ; 3. Le système effectue la somme des facturations de chaque discussion ; 4. Le système enregistre la facture comme étant une créance du client.
❖ Scénario d'exceptions
<ul style="list-style-type: none"> • E1 : Il n'a aucune prestation à facturer ✓ Ce scénario débute au point 1 du scénario nominal; ✓ Le scénario nominal est interrompu.
❖ Scénario alternatif
<ul style="list-style-type: none"> • A1 : L'agent envoie une salutation sous forme de notification au client ✓ Il débute à l'étape 1 du scénario nominal;
❖ Post conditions
<ul style="list-style-type: none"> ✓ La facture est générée ; ✓ Le journal est mis à jour.

- Réclamer paiement

Tableau 10: description textuelle du cas d'utilisations réclamer paiement

Sommaire d'identification
<p><u>Titre</u> : Réclamer paiement</p> <p>Acteurs : Agents</p> <p><u>Résumé</u> : Consiste à attirer l'attention d'un client sur une facture non payé</p> <p><u>Responsable</u> : ADUAYOM MESSAN E. Enoc</p> <p>Version : 1.0</p> <p><u>Date</u> : 04/08/2017</p>
Description des scénarii
❖ Pré condition
<p>✓ Le système est fonctionnel ;</p> <p>✓ L'utilisateur est connecté.</p>
❖ Scénario nominal
<p>1. L'agent consulte les créances ;</p> <p>2. L'agent choisit la/les facture(s) concernée(s) ;</p> <p>3. L'agent lance la réclamation ;</p> <p>4. Le système envoie une notification au client.</p>
❖ Scénario d'exceptions
<ul style="list-style-type: none"> • E1 :Le client est à jours (aucune créance)

- ✓ Ce scénario débute au point 1 du scénario nominal;
- ✓ Le système affiche à l'agent qu'il n'y a aucune créance;
- ✓ Le scénario nominal est interrompu.

❖ Scénario alternatif

- A1 : L'agent génère une facture
- ✓ Il débute à l'étape 1 du scénario nominal;
- ✓ L'agent génère une facture (Tableau 6);
- ✓ Le scénario nominal recommence.

❖ Post conditions

- ✓ Le paiement est réclamé ;
- ✓ Le journal est mis à jour.

- Réclamer facture

Tableau 11: description textuelle du cas d'utilisation réclamer facture

Sommaire d'identification
<p><u>Titre</u> : Réclamer facture</p> <p>Acteurs : Client</p> <p><u>Résumé</u> : Consiste à demander à l'agent qu'on lui génère ultérieurement une facture</p> <p><u>Responsable</u> : ADUAYOM MESSAN E. Enoc</p> <p>Version : 1.0</p>

Date : 04/08/2017
Description des scénarii
❖ Pré condition
<ul style="list-style-type: none"> ✓ Le système est fonctionnel ; ✓ L'utilisateur est connecté.
❖ Scénario nominal
<ol style="list-style-type: none"> 1. Le client consulte les prestations aux quels il a eu droit ; 2. Le client choisi la/les prestations(s) concernée(s) ; 3. Le client lance la demande ; 4. Le système envoi une notification à l'agent.
❖ Scénario d'exceptions
<ul style="list-style-type: none"> • E1 :Il n'y a aucune prestation à facturer ✓ Ce scénario débute au point 1 du scénario nominal; ✓ Le scénario nominal est interrompu.
❖ Scénario alternatif
<ul style="list-style-type: none"> • A1 :Le client annule la demande ✓ Ce scénario débute au point 3 du scénario nominal; ✓ Le scénario nominal est interrompu.
❖ Post conditions
<ul style="list-style-type: none"> ✓ La facture est réclamé ;

✓ Le journal est mis à jour.

- Ajout d'un utilisateur

Tableau 12: description textuelle du cas d'utilisations ajouter utilisateur

Sommaire d'identification
<p><u>Titre</u> : Ajouter utilisateur</p> <p>Acteurs : Agents</p> <p><u>Résumé</u> : Consiste à enregistrer un nouveau utilisateur</p> <p><u>Responsable</u> : ADUAYOM MESSAN E. Enoc</p> <p>Version : 1.0</p> <p><u>Date</u> : 04/08/2017</p>
Description des scénarii
❖ Pré condition
<p>✓ Le système est fonctionnel ;</p> <p>✓ L'utilisateur est connecté.</p>
❖ Scénario nominal
<ol style="list-style-type: none"> 1. L'agent choisit le type d'utilisateur à ajouter; 2. L'agent saisie les informations ; 3. le système vérifie les informations saisies ; 4. Le système enregistre l'utilisateur.
❖ Scénario d'exceptions

<ul style="list-style-type: none"> • E1 :Le client existe déjà ✓ Ce scénario débute au point 3 du scénario nominal; ✓ Le système lui redemande de vérifier les informations saisie; ✓ Le scénario nominal est continu.
❖ Scénario alternatif
<ul style="list-style-type: none"> • A1 :L'ajout est annulé ✓ Ce scénario débute au point 2 du scénario nominal; ✓ Le scénario nominal est interrompu.
❖ Post conditions
<ul style="list-style-type: none"> ✓ L'enregistrement est effectué ; ✓ Le journal est mis à jour.

- Suspendre client

Tableau 13:description textuelle de cas d'utilisations de suspendre un client

Sommaire d'identification
<p>Titre : Suspendre client</p> <p>Acteurs : Agents</p> <p>Résumé : Consiste à restreindre l'accès aux discussions à un client</p> <p>Responsable : ADUAYOM MESSAN E. Enoc</p> <p>Version : 1.0</p>

Date : 04/08/2017
Description des scénarii
❖ Pré condition
<ul style="list-style-type: none"> ✓ Le système est fonctionnel ; ✓ L'utilisateur est connecté.
❖ Scénario nominal
<ol style="list-style-type: none"> 1. L'agent choisit le client dont il est question; 2. L'agent lance la suspension ; 3. le système vérifie l'état des créances du client ; 4. L'agent saisie le motif de suspension.
❖ Scénario d'exceptions
<ul style="list-style-type: none"> • E1 :Le client est à jours ✓ Ce scénario débute au point 3 du scénario nominal; ✓ Le scénario nominal est continu.
❖ Scénario alternatif
<ul style="list-style-type: none"> • A1 :La suspension est annulé ✓ Ce scénario débute au point 2 du scénario nominal; ✓ Le scénario nominal est interrompu.
❖ Post conditions
<ul style="list-style-type: none"> ✓ La suspension est effectué ;

✓ Le journal est mis à jour.

2. Diagrammes de séquence

Ils permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs.

- Paiement

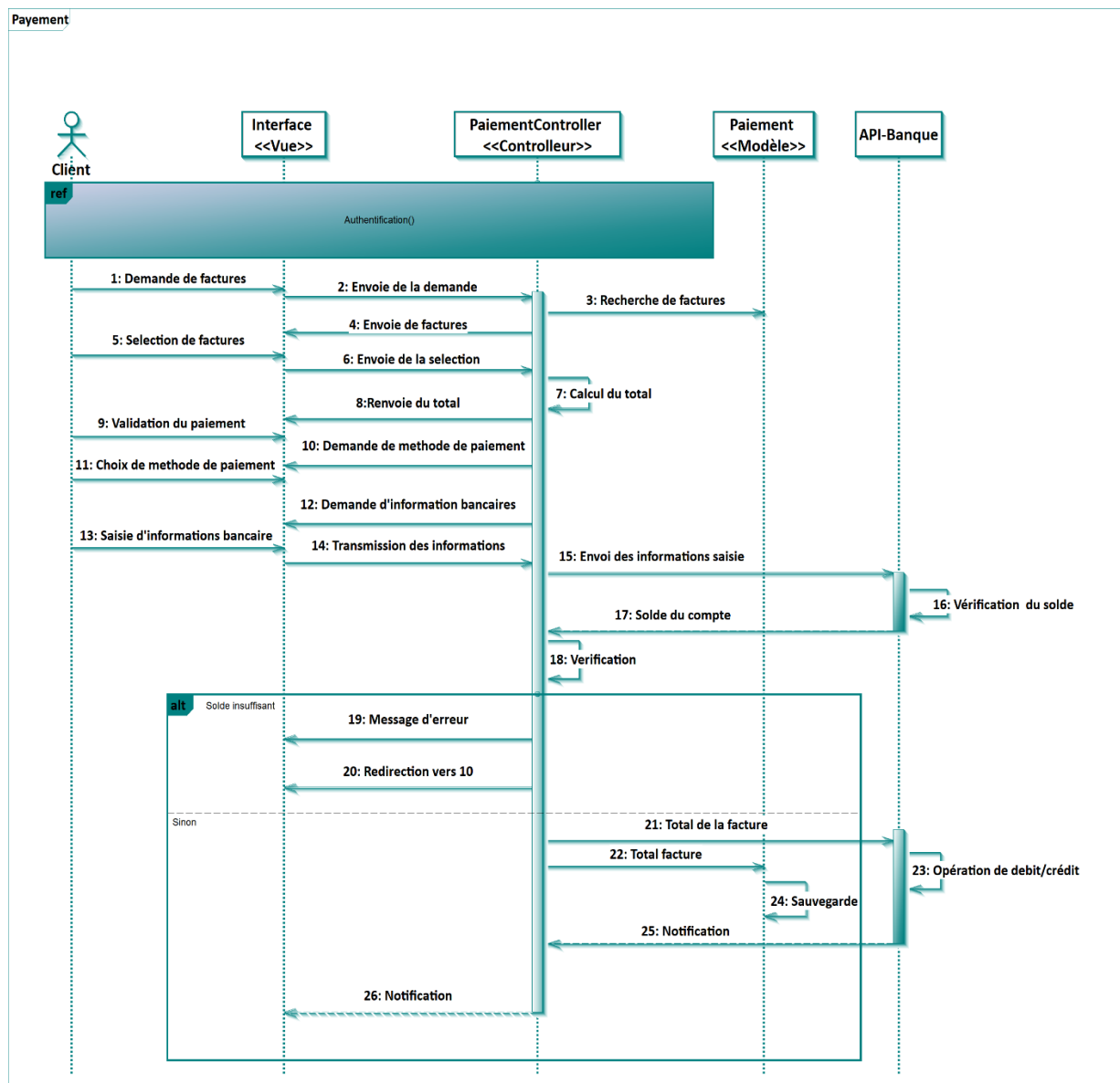


Figure 13: diagramme de séquences de paiement

- Discussion

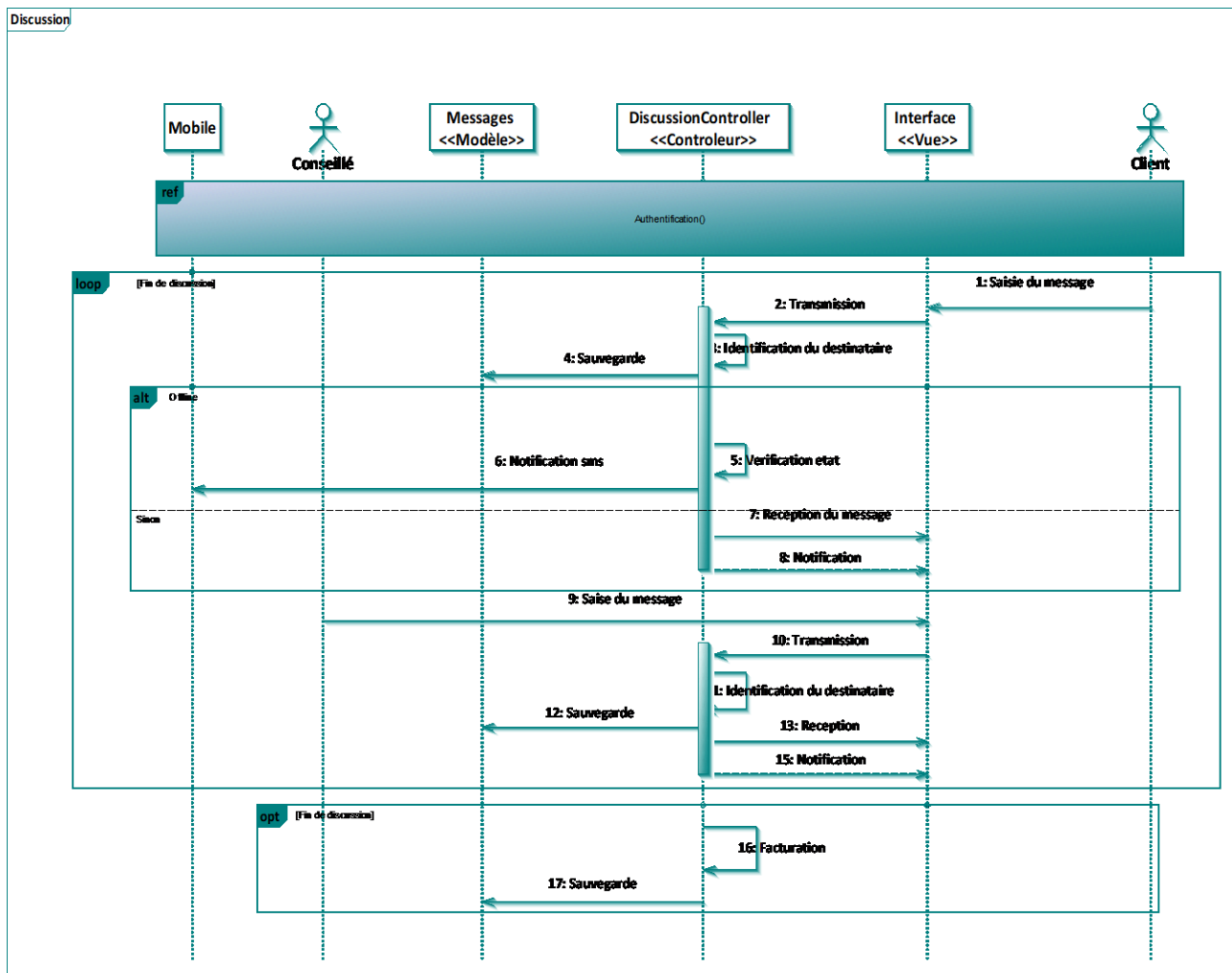


Figure 14: diagramme de séquences de discussion

- Ajout utilisateur

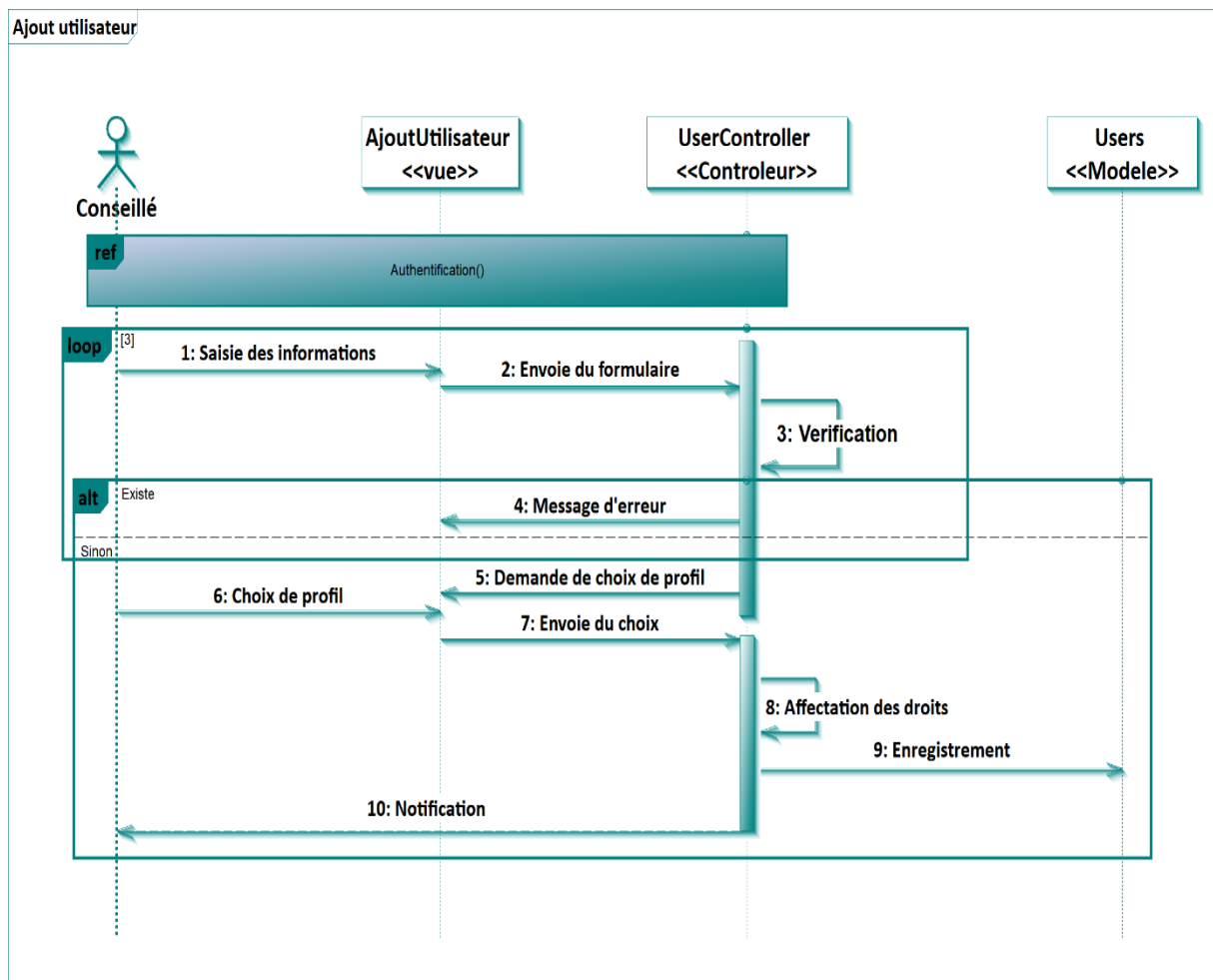


Figure 15: diagramme de séquences d'ajout d'utilisateur

- Authentification

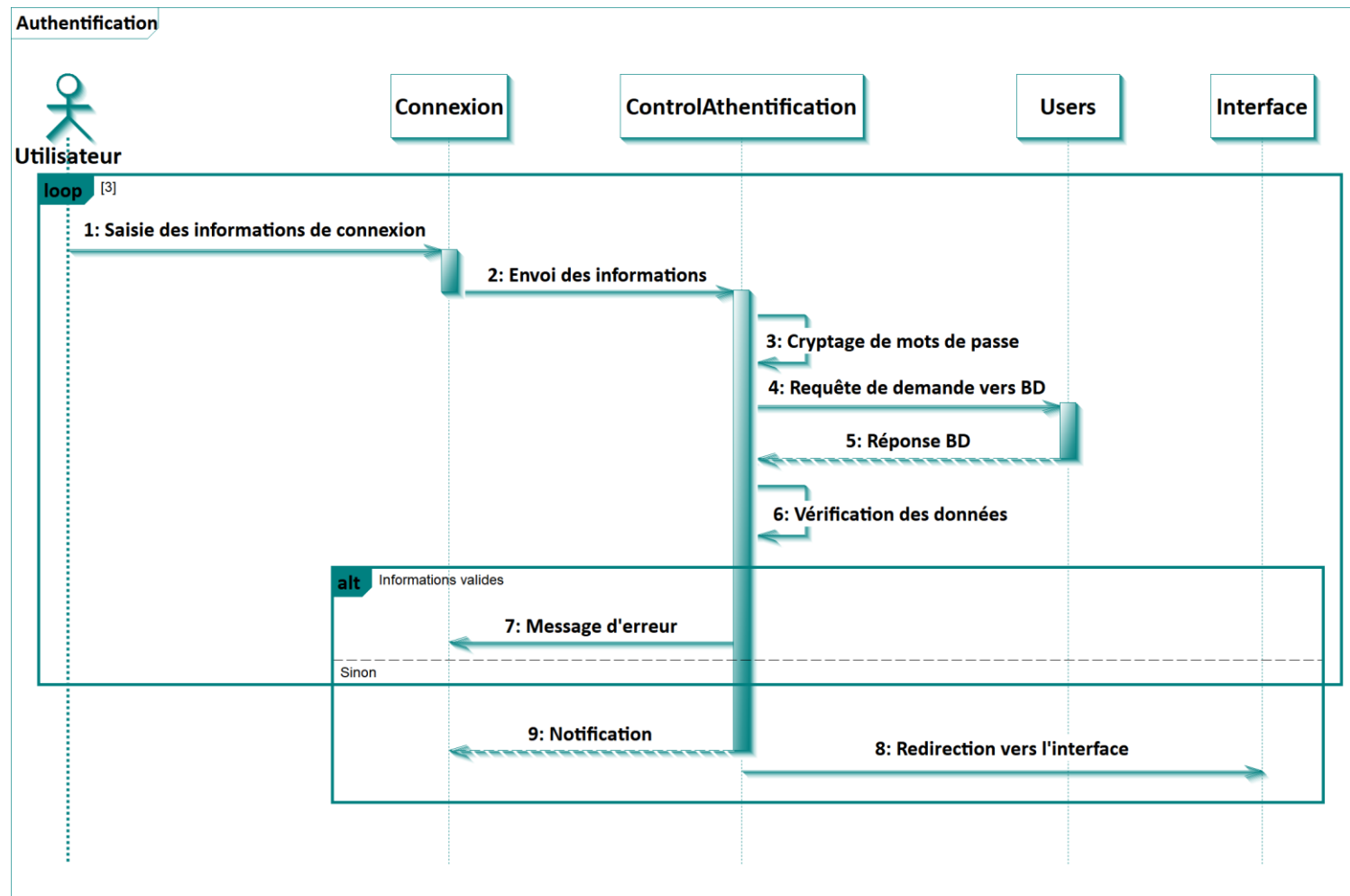


Figure 16: diagramme de séquences d'authentification

3. Diagrammes d'activités

Le diagramme d'activités permet 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.

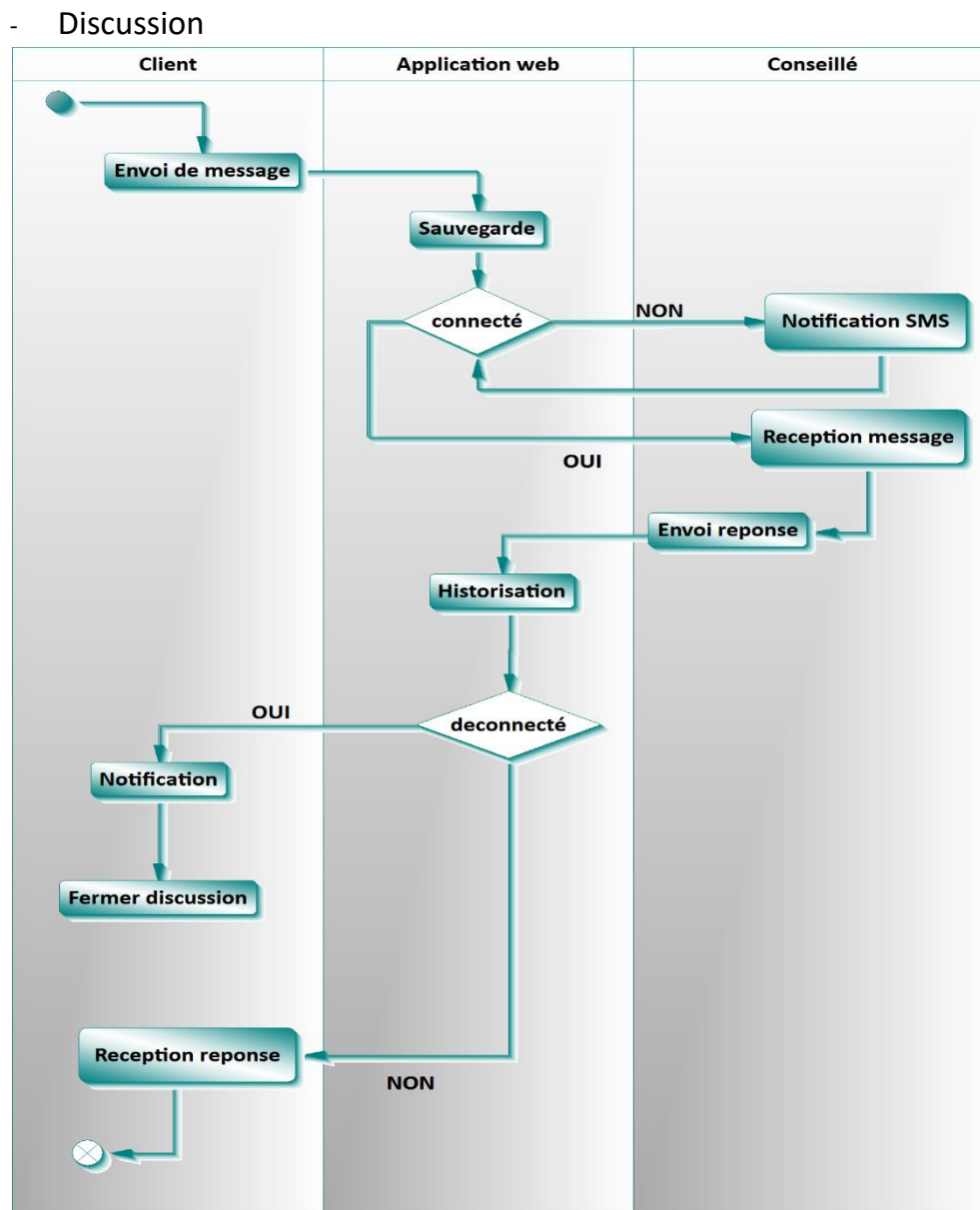


Figure 17: diagramme d'activités de discussion

- Payement

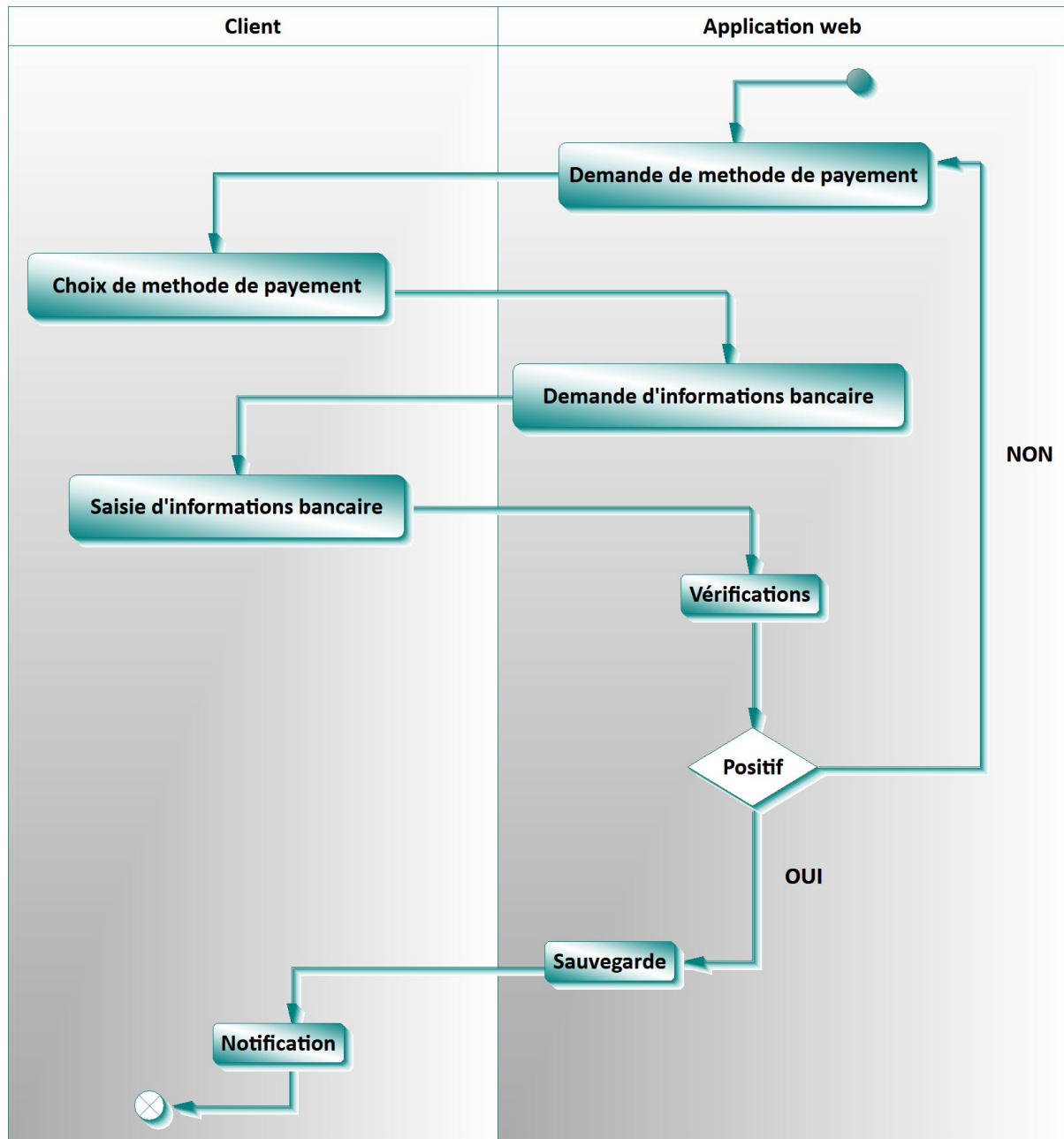


Figure 18: diagramme d'activités de paiement

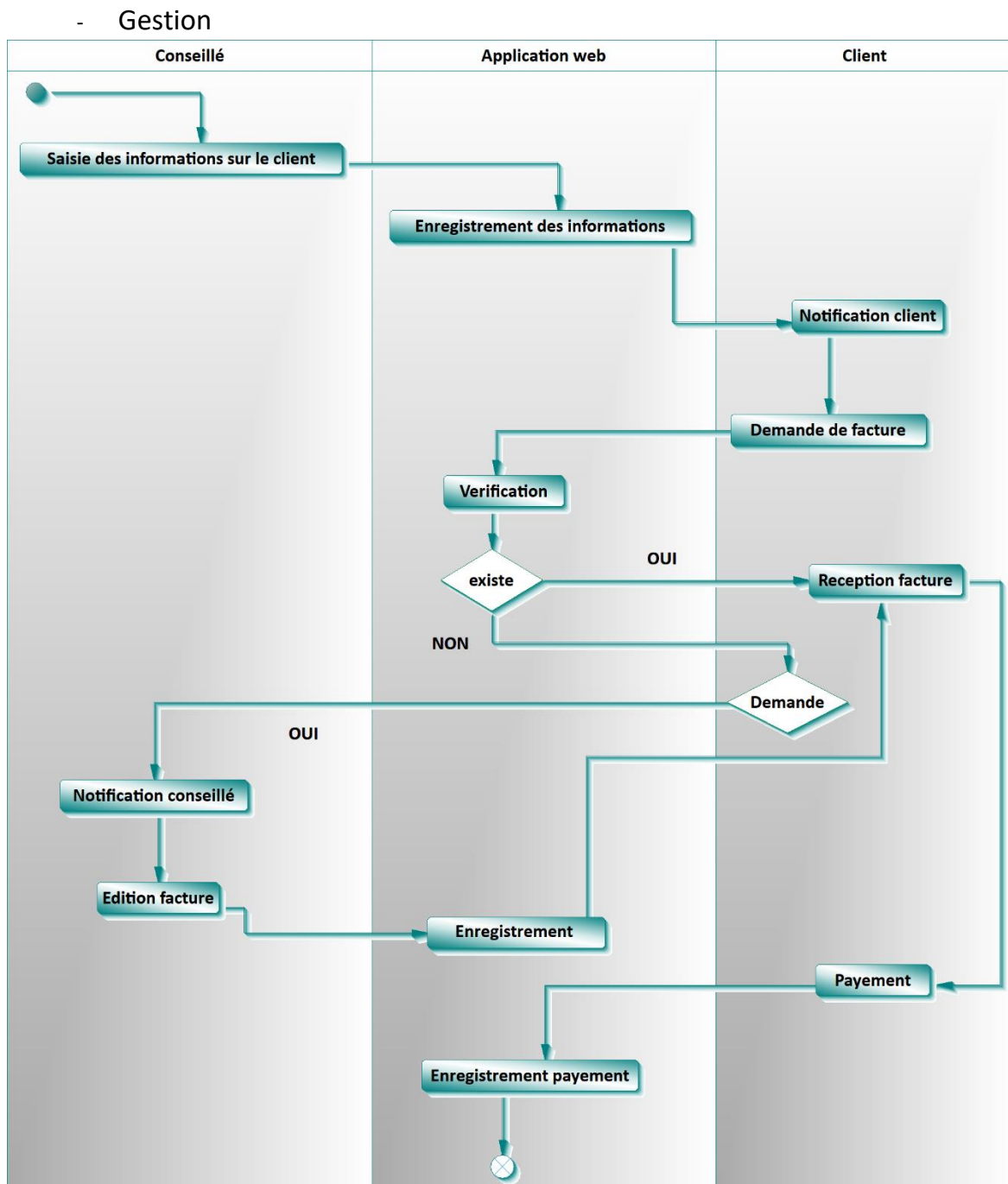


Figure 19: diagramme d'activités de gestion

- Authentification

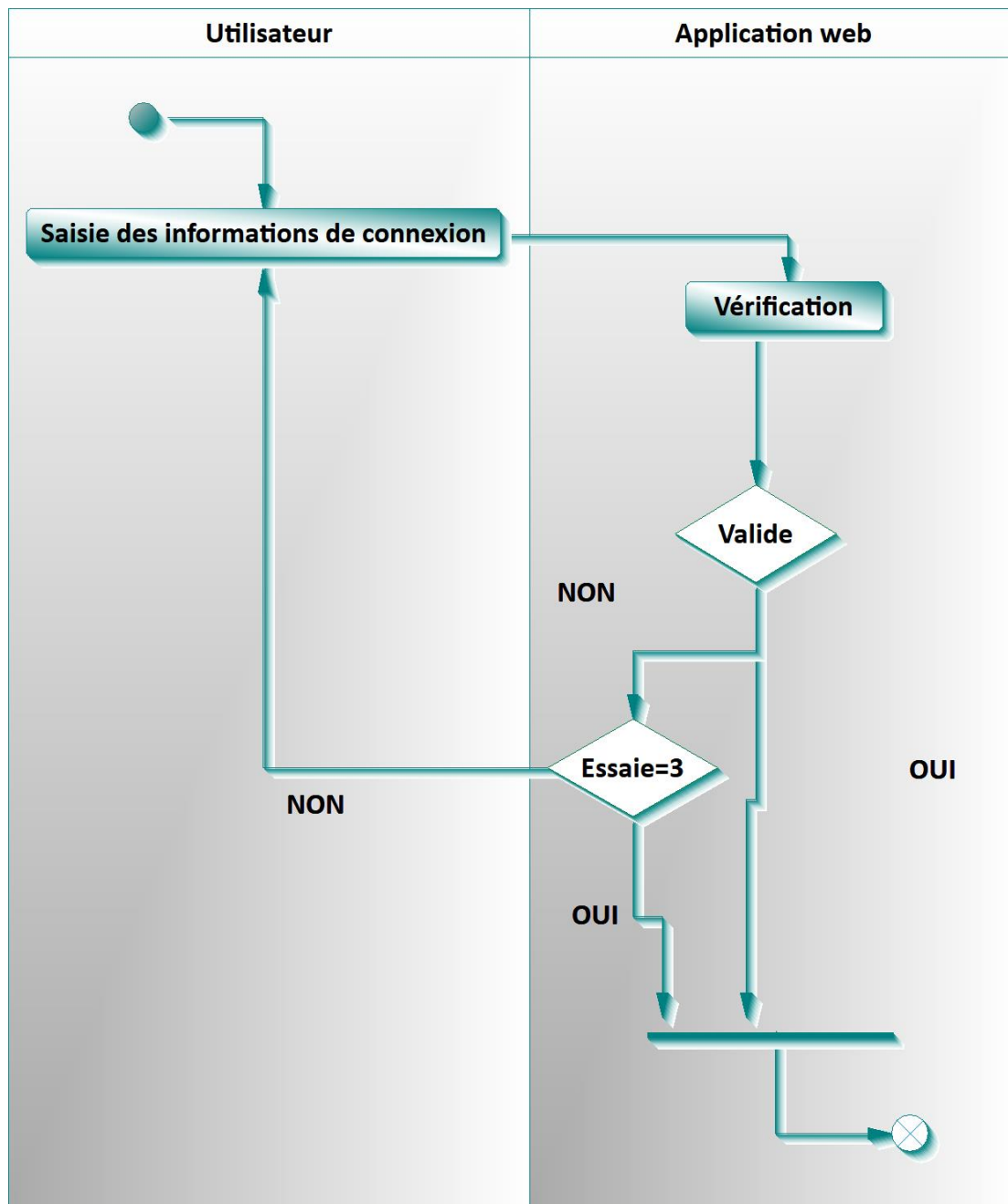


Figure 20: diagramme d'activités d'authentification

4. Diagrammes de classe

Ils permettent de spécifier la structure et les liens entre les objets dont le système est composé. Ils spécifient QUI sera à l'œuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation.

- Gestion de messagerie

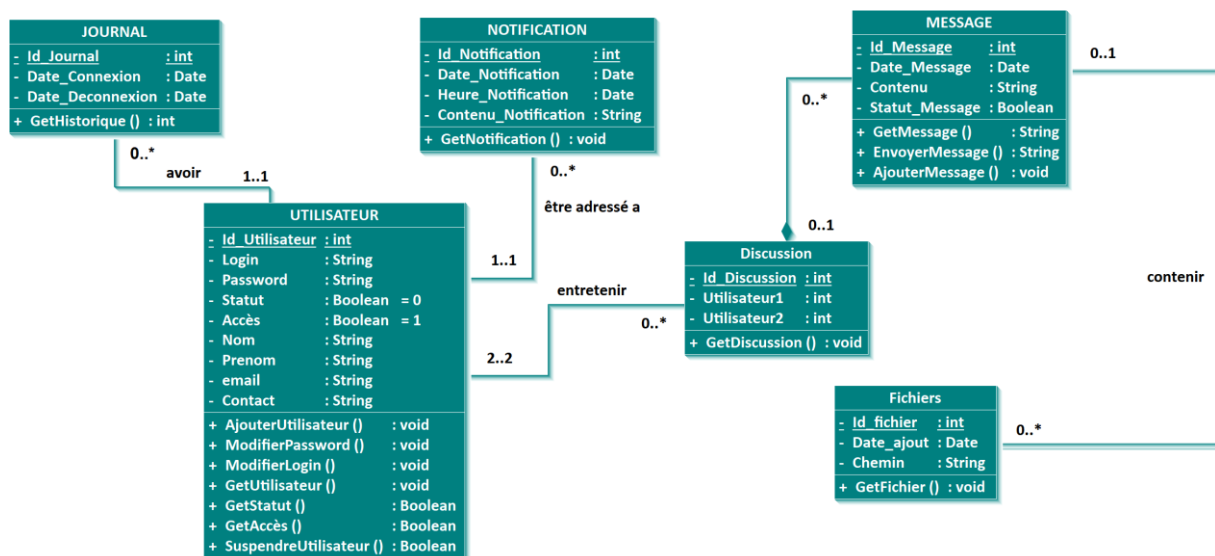


Figure 21:gestion des messages

- Administration

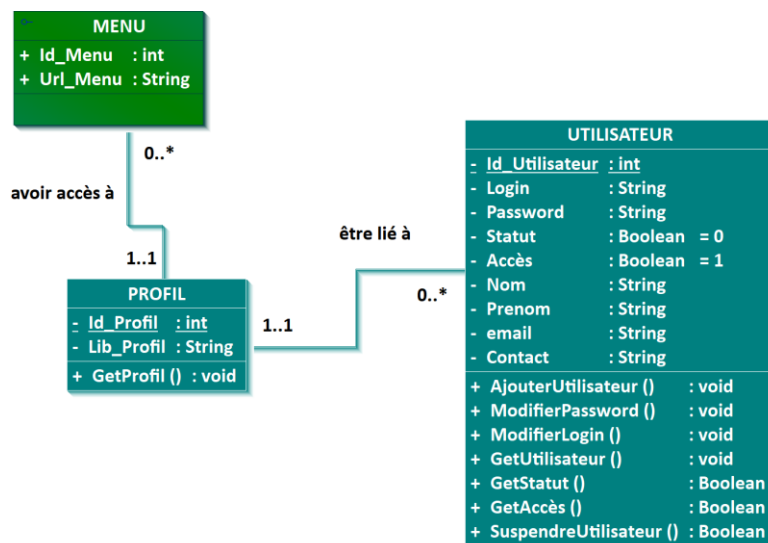


Figure 22:diagramme de classe (administration)

- Gestion clients

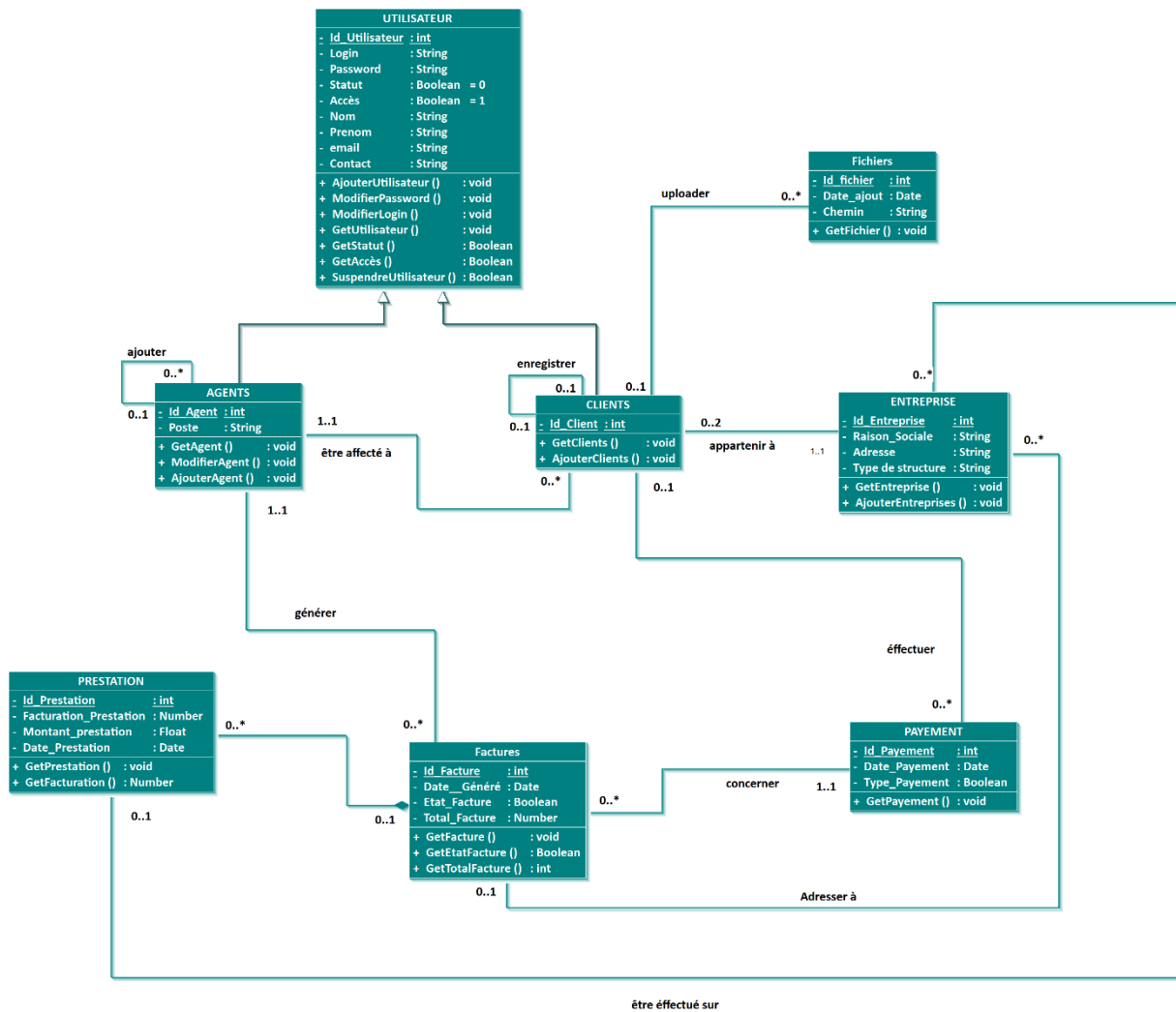


Figure 23:gestion de facture

- Diagramme général

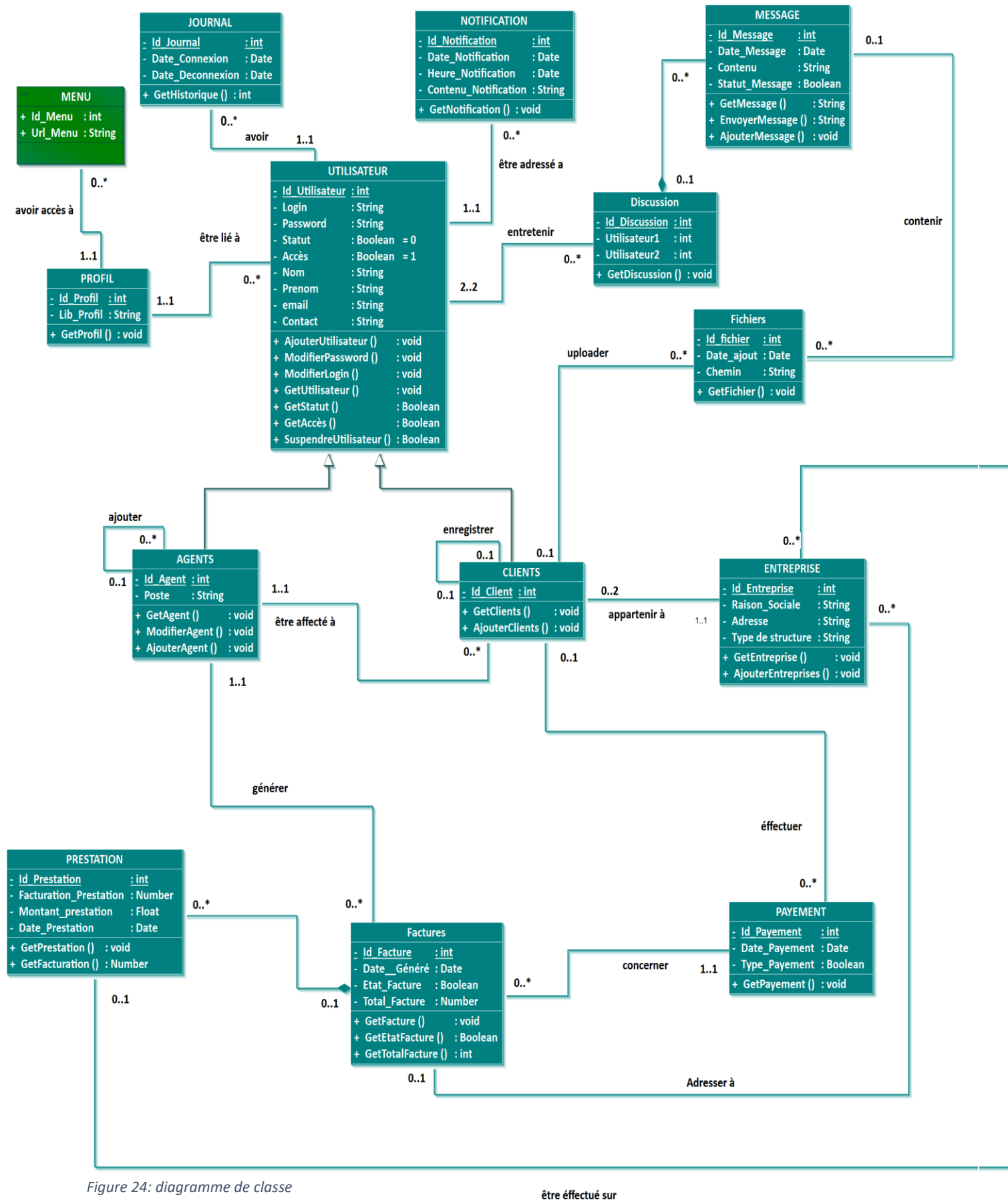


Figure 24: diagramme de classe

être effectué sur

CONCLUSION

Nous venons de terminer la phase d'analyse et de conception. Après avoir posé les bases nous allons entamer naturellement la phase de mise en œuvre et de réalisation où nous exposerons les différents outils utilisés pour la mise en œuvre de notre application.

PARTIE III

MISE EN ŒUVRE ET REALISATION

Du 29 Août 2017 Au 15 Septembre 2017

INTRODUCTION

Suite à la présentation du cahier des charges et du dossier de conception nous avons pu déterminer les données utilisées pour notre projet et les traitements qui y sont effectués.

Nous abordons à présent le dossier de mise en œuvre et de réalisation qui constitue le troisième document de notre mémoire.

Ce dossier comprend deux volets que sont la mise en œuvre et la présentation de l'application.

I. MISE EN ŒUVRE

1. Matériels et logiciels utilisés

1.1. Choix matériels

Pour que notre système fonctionne le plus optimal possible nous avons besoins des matériels suivant :

Tableau 14: récapitulation de matériels choisis

Matériels	CARACTERISTIQUES
Carte SIM (pour test)	-
Modem professionnel	-
Ordinateur Ultrabook	Core i7 ,RAM 32 giga, HDD 500 giga
Téléphone Android	QUADCORE, OS 6.0, RAM 3giga, APN 16 MP
Serveur dédié	Processeur Intel Xeon Core i7 3.5GHz, RAM 32Go, Disque 512Go SSD + 4*500Go HDD

1.2. Choix logiciels

i. Outils et technologies de développement

- **Laravel**



Figure 25: logo LARAVEL

Laravel, le logiciel que nous allons utiliser pour gérer notre système est distribué sous licence MIT avec ses sources hébergées sur Git hub. Il est un Framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT avec ses sources hébergées sur Git hub.

La cinquième version de Laravel est écrite en PHP 5 et son installation est basée sur le gestionnaire de paquets Composer. Laravel fournit des fonctionnalités en termes de routage de requêtes, de mapping objet-relationnel (un système baptisé Eloquent implémentant Active Record), d'authentification, de vue (avec Blade), de migration de base de données, de gestion des exceptions et de tests unitaires. Pour la réalisation de la plateforme web de notre application, nous avons opté pour l'utilisation du Framework Laravel.

- **NOWSMS**

NowSms est une solution pour l'envoi et la réception des SMS ou de MMS via un modem GSM connecté à un pc via un port USB.

Les fonctions et avantages de Nowsms :

- ❖ Permettre l'envoi des MMS et des SMS ;
- ❖ Utilisable sous Windows et sur des serveurs Windows ;
- ❖ Utilise un ou plusieurs modems GSM ou n'importe quelle connexion TCP/IP utilisant des protocoles comme SMPP, UCP/EMI et ou HTTP du côté du client et des protocoles comme SMTP, SMPP, HTTP du côté serveur ;
- ❖ Notons tout de même que Nowsms ne remplace en aucun cas les services de messagerie ;

- ❖ Peut s'utiliser avec internet (d'où les protocoles SMPP, HTTP...) ou dans un réseau local ou privé.

- **GIT**



Figure 26:logo GIT

Git est un logiciel libre de gestion de versions décentralisé créé par Linus Torvalds auteur du noyau Linux et distribué selon les termes de la licence publique générale GNU version 2. En 2016. Git est un logiciel de gestion de versions le plus populaire et utilisé par plus de douze millions de personnes.

Il nous a permis de mettre en place le Versioning des codes sources et documents de notre application. Pour son implémentation sur le web, nous avons utilisé Bitbucket plateforme présentée à la suite.

- **BITBUCKET**



Figure 27:logo BITBUCKET

Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant les logiciels de gestion de versions Git (logiciel de gestion de versions décentralisé) et Mercurial (logiciel de gestion de versions décentralisé disponible) par l'intermédiaire de logiciels comme SourceTree.

Bitbucket est une plateforme web qui permet d'effectuer le clouding. Un petit rappel, au sein de notre entreprise nous avons décidé d'opter pour la méthode AGILE ce qui implique un travail de groupe. Bitbucket nous permet alors de pouvoir travailler quel que soit la distance de suivre l'évolution de nos codes. Il nous permet également d'effectuer le Versionning et de faciliter le suivi et le développement du projet en groupe.

- **PHP Storm (Plateforme de développement)**



Figure 28: logo PHP Storm

PHP Storm est un éditeur pour PHP, HTML et JavaScript, édité par JetBrains. Nous l'avons spécialement choisi parce qu'il nous permet d'intégrer le Versionning. Ses avantages sont multiples. Il propose :

- Une coloration syntaxique ;
- Affichage des erreurs à la volée ;
- Auto-complétion intelligente du code ;
- Ré usinage du code ;
- L'envoi des fichiers via FTP pour le déploiement ;
- Un gestionnaire de version.

Il permet aussi de visualiser l'architecture de bases de données de différentes sources (MySQL, SQLite, ...).

- **MySQL (Base de données)**



Figure 29: logo MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. MySQL fait partie des logiciels de gestion de base de données les plus utilisés au monde autant par le grand public (applications web principalement) que par des professionnels en concurrence avec Oracle, Informix et Microsoft SQL Server.

Acheter le 16 janvier 2008 par Sun Microsystems, MySQL tient son nom du prénom de la fille du Co créateur Michael Widenius. MySQL faisant référence au Structured Query Language, le langage de requête utilisé.

Depuis mai 2009, Michael Widenius créateur de MySQL a créé Maria DB pour continuer son développement en tant que projet Open Source.

- **Socket**



Figure 30: logo Socket

Dans le contexte des logiciels, on peut le traduire par « connecteur réseau » ou « interface de connexion ».

Apparu dans les systèmes UNIX, un socket est un élément logiciel qui est aujourd'hui répandu dans la plupart des systèmes d'exploitation. Il s'agit d'une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau.

Il lui sera ainsi aisé d'établir une session TCP, puis de recevoir et d'expédier des données grâce à elle. Cela simplifie sa tâche car cette couche logicielle de laquelle il requiert des services en appelant des fonctions, masque le travail nécessaire de gestion du réseau pris en charge par le système. Le terme socket désigne en pratique chaque variable employée dans un programme afin de gérer l'une des sessions.

La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication :

- Le mode connecté (comparable à une communication téléphonique) utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus de telle façon que l'adresse de destination ne soit pas nécessaire à chaque envoi de données ;
- Le mode non connecté (analogue à une communication par courrier) utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi et aucun accusé de réception n'est donné.

Les sockets sont implémentés dans différents langages (C, Java, ...). En langage C, elles utilisent des fonctions et des structures disponibles dans la bibliothèque <sys/socket.h>.

- **Node.js**



Figure 31:logo Node.js

Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.

Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS.

Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui permet de faire tourner un serveur web sans avoir besoin d'un logiciel externe comme Apache ou lighttpd, et de mieux contrôler la façon dont le serveur web fonctionne.

Concrètement, node.js est un environnement d'assez bas niveau permettant d'exécuter du JavaScript non plus dans le navigateur web mais sur le serveur.

Node.js est de plus en plus populaire comme plateforme serveur.

- **Mustache**



Figure 32: Mustache

Mustache est un moteur de Template avec des implémentations disponibles en ActionScript, C++, Clojure, Coffee Script, Cold Fusion, Delphi, Erlang(langage), Go, Haskell, Io, Java, JavaScript, Lua, .NET, ObjectiveC, Pharo, Perl, PHP, Python, Ruby, Scala, CFEngine et XQuery.

- **Material Design**



Figure 33: logo material design

Le Material Design est un ensemble de règles de design proposées par Google et qui s'appliquent à l'interface graphique des logiciels et applications. Il est utilisé notamment à partir de la version 5.0 du système d'exploitation Android.

Google a présenté le Material Design pour la première fois lors de la conférence Google I/O, le 25 juin 2014. En misant sur les motifs « carte » déjà utilisés dans Google Now, ces règles de design mettent l'accent sur une utilisation accrue des mises en page basées sur une grille des animations et des transitions, des effets de profondeur tels que l'éclairage et les ombres. Selon Google ce nouveau langage de design est basé sur le papier et l'encre.

Le designer Matias Duarte explique que « contrairement au vrai papier, notre matériau numérique peut s'étirer et se modifier de manière intelligente. Le matériau contextuel

a une surface physique et des bords. Les superpositions et les ombres donnent des informations sur ce que vous pouvez toucher ».

Ce design d'interface utilisateur succède au design Holo utilisé par Android de la version 4.0 à la version 4.4.

- **JQuery**



Figure 34: logo JQuery

jQuery est une bibliothèque JavaScript multiplateforme conçue pour simplifier les scripts côté HTML du HTML. C'est un logiciel open source gratuit utilisant la licence MIT permissive. L'analyse Web indique que c'est la bibliothèque JavaScript la plus largement déployée d'une large marge.

La syntaxe de jQuery est conçue pour faciliter la navigation dans un document, sélectionner les éléments DOM, créer des animations, gérer des événements et développer des applications Ajax. jQuery fournit également des fonctionnalités aux développeurs pour créer des plug-ins sur la bibliothèque JavaScript. Cela permet aux développeurs de créer des abstractions pour une interaction et une animation de bas niveau, des effets avancés et des widgets à haut niveau et à thème. L'approche modulaire de la bibliothèque jQuery permet la création de puissantes pages web et applications Web dynamiques.

L'ensemble des fonctionnalités de base de jQuery-sélections d'éléments DOM, traversée et manipulation activés par son moteur de sélection (appelé "Sizzle" de v1.3) a créé un nouveau "style de programmation", des algorithmes de fusion et des structures de données DOM. Ce style a influencé l'architecture d'autres Framework JavaScript comme YUI v3 et Dojo, stimulant plus tard la création de l'API standard Selectors.

2. Architecture matérielle et logicielle de l'application

- Architecture matérielle

Description :

- Les utilisateurs qui effectuent des actions sur les différentes interfaces
- Ces interfaces qui communiquent à l'aide d'un protocole directement ou à l'aide d'un web service avec la base de données.

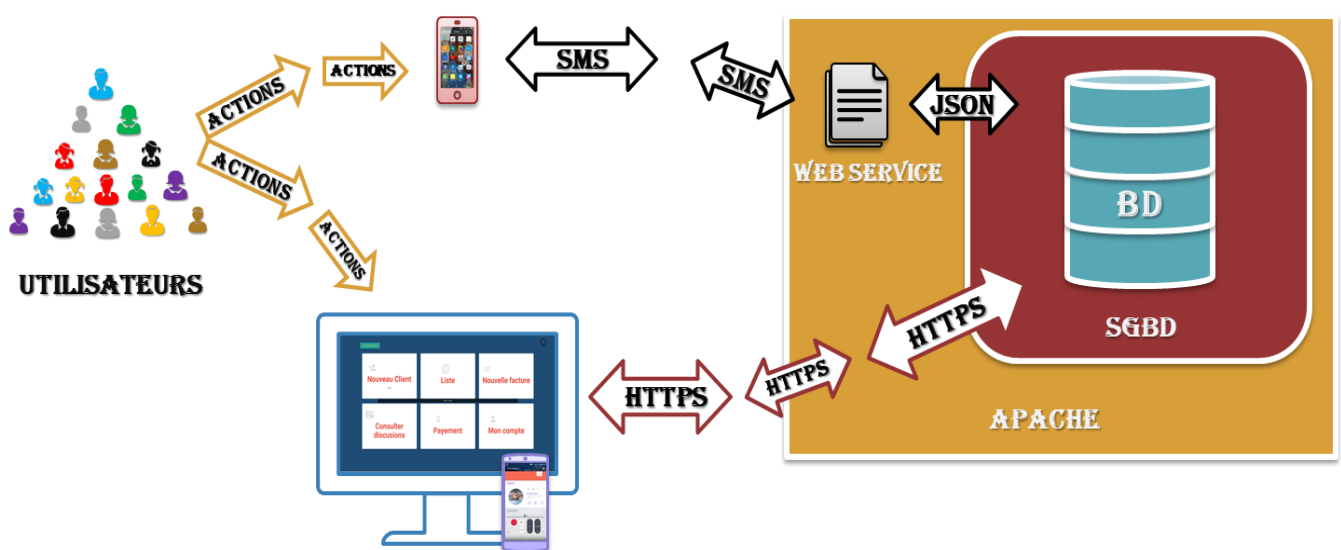


Figure 35: architecture de l'application

- Architecture logicielle

Pour notre application nous avons appliqué une architecture de type MVC (Modèle Vue Contrôleur).

Modèle-vue-contrôleur ou **MVC** est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

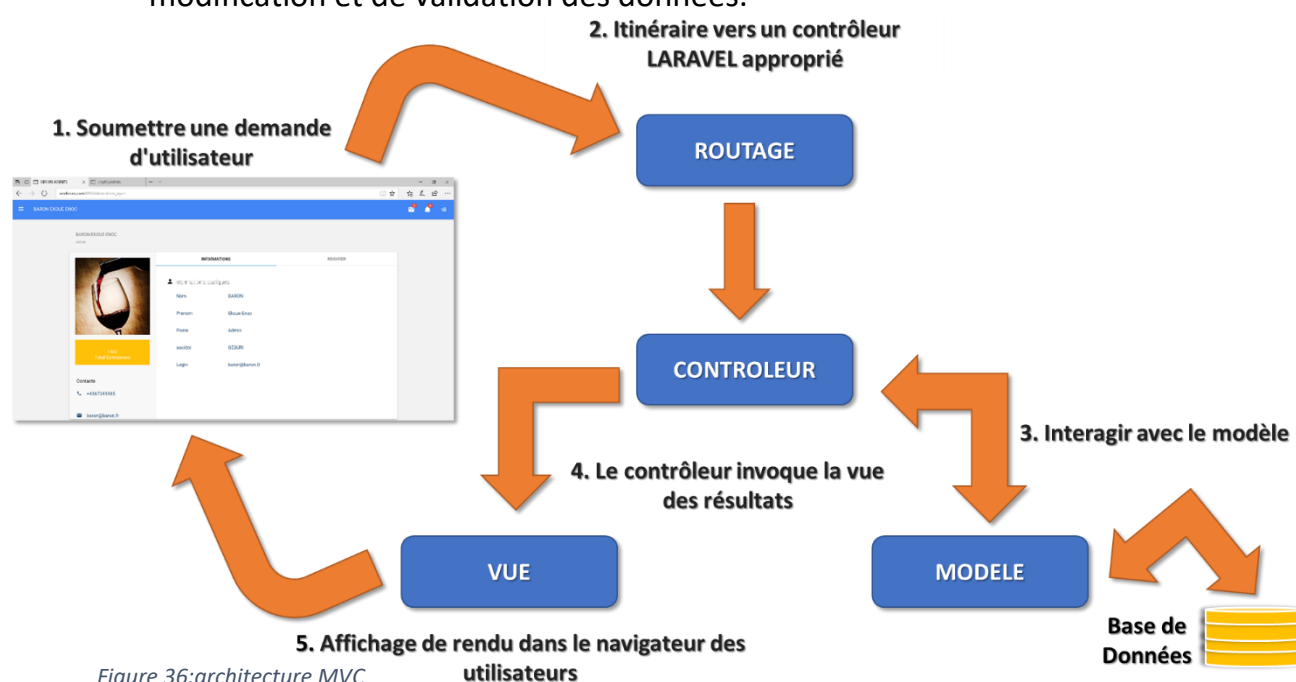
- Un modèle (Model) contient les données à afficher.
- Une vue (View) contient la présentation de l'interface graphique.
- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

Une application conforme au motif MVC comporte trois types de modules : les modèles, les vues et les contrôleurs.

- **Modèle** : Élément qui contient les données ainsi que de la logique en rapport avec les données (validation, lecture et enregistrement). Il peut, dans sa forme la plus simple, contenir uniquement un simple texte, voire des données beaucoup plus compliquées. Le modèle représente l'univers dans lequel s'inscrit l'application. Par exemple pour une application de banque, le modèle représente des comptes, des clients, ainsi que les opérations telles que dépôt et retraits, et vérifie que les retraits ne dépassent pas la limite de crédit. Le modèle est indépendant de la vue et du contrôleur et ne s'en sert pas.
- **Vue** : Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle³. Dans une application web une vue contient des balises HTML.
- **Contrôleur** : Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

Les trois éléments sont indépendants les uns des autres, le modèle ne se sert ni de la vue ni du contrôleur, il peut cependant leur envoyer des messages. Il y a deux liens entre la vue et le modèle (premièrement la vue lit les données du modèle et deuxièmement reçoit des messages provenant du modèle). Dans la mesure où une vue est associée à un modèle et un modèle est indépendant, un même modèle peut être utilisé par plusieurs vues.

- Le contrôleur dépend de la vue et du modèle : la vue comporte des éléments visuels que l'utilisateur peut actionner. Le contrôleur répond aux actions effectuées sur la vue et modifie les données du modèle.
- Dans le cas d'un view model, le modèle contient les données que le contrôleur transmet à la vue. Dans le cas d'un Domain model il contient toutes les données en rapport avec l'activité, ainsi que la logique des opérations de modification et de validation des données.



3. Déploiement de l'application

Nous avons déployé notre système sur un serveur web Apache. La partie suivante décrit les étapes suivies pour le déploiement.

- Nous avons cloné le projet qui était hébergé sur une plateforme git (Bitbucket) ;
- Nous avons importé sur le serveur les tables de notre système dans une base de données créées avec Phpmyadmin ;
- Nous avons adapté le fichier d'extension .env du système ;

- Nous avons édité le fichier de configuration du site pour personnaliser le fonctionnement de l'application ;
- Nous avons activé SSL (Secure Socket Layer) avec la commande « a2nmode ssl ». Il permet d'échanger des informations entre 2 ordinateurs de façon sûre ;
- Nous avons activé le site avec La commande « a2ensite » ;
- Nous avons rechargé le serveur apache pour prendre en compte les modifications avec la commande « service apache2 reload ».

La phase de déploiement est terminée. Nous avons maintenant accès au site web.

4. Sécurité de l'application

- Une gestion de l'historique très ferme. N'oublions pas de même la gestion de la sécurité soit de privilèges et de niveaux d'accès pour les différents types d'utilisateurs (Agents, administrateurs et Clients). Selon leur statut, le contenu des pages varie et l'accès aux informations avec un statut supérieur est interdit. L'identification des différents intervenants devra s'exécuter par un service central d'authentification.
- Chaque opération qui sera effectuée par le client sera automatiquement sauvegardée. On pourra suivre à la trace les activités du client pour permettre une gestion plus aiguisée des clients.
- Auront accès au système seuls les ayant accès, un service d'authentification est mis en place à cet effet.
- L'application est elle-même étant hébergée sur un serveur web distant, les données sensibles seront sécurisées par des protocoles de Hachages comme le md5.
- La communication entre les différents serveurs, services et parties du système se fera par le protocole HTTPS pour garantir la sécurité des transactions et palier aux risques de piratage.

II. PRÉSENTATION DE L'APPLICATION

1. Présentation

1.1 Maquette

Voici comment se présente notre application :

Informations des connectés	En tête de page	Icônes de notification
Contenu de la page		Liste des Connectés
Pied de page		

Figure 37: maquette de l'application

1.2 Structure

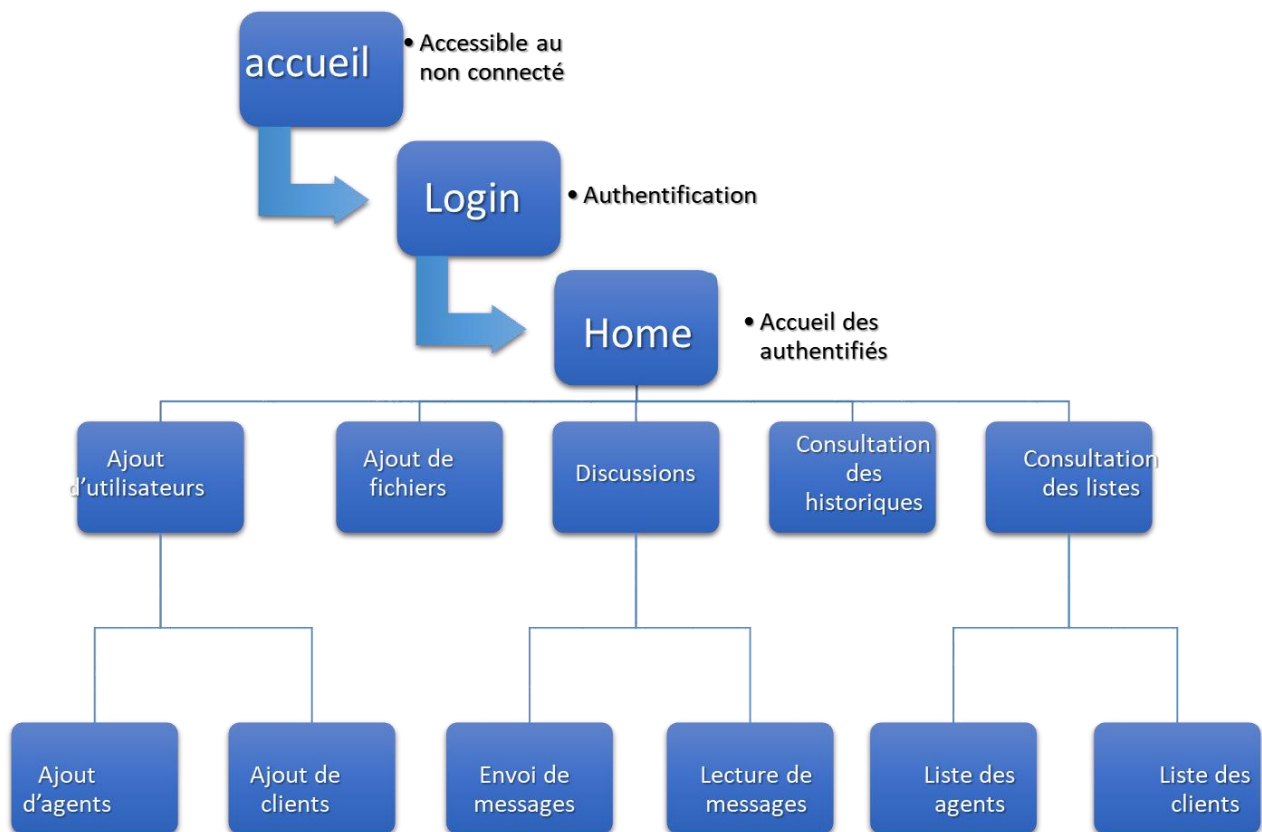


Figure 38: structure de l'application

1.3 Description textuelle

Notre solution est une application Web de gestion de clients et de discussion instantanée.

À l'accueil, un utilisateur non enregistré a la possibilité de discuter avec les agents du cabinet.

Après authentification, l'utilisateur a alors accès à un menu qui lui permettra selon ses besoins et ses droits d'accès d'effectuer diverses actions. Le menu auquel il aura accès suite à l'authentification est un centenaire d'actions, l'utilisateur peut alors à sa guise effectuer des opérations.

2. Script de création de la base de données

```

/*=====*/
/* Table: Discussion */
/*=====*/
create table Discussion
(
    idDiscussion    int not null,
    utilisateur1    int,
    utilisateur2    int,
    primary key (idDiscussion)
);

/*=====*/
/* Table: Entreprise */
/*=====*/
create table Entreprise
(
    idEntreprise    int not null,
    idFacture        int,
    Pre_idFacture    int,
    idPrestation     int,
    raisonSociale    varchar(35),
    adresse          varchar(35),
    typeDeStructure  varchar(25),
    primary key (idEntreprise)
);

/*=====*/
/* Table: Factures */
/*=====*/
create table Factures
(
    idFacture        int not null,
    idPayement        int not null,
    idUtilisateur     int not null,
    idAgent           int not null,
    dateGenere        datetime,
    etatFacture        bool,
    totalFacture       numeric(8,0),
    primary key (idFacture)
);

/*=====*/
/* Table: Journal */
/*=====*/
create table Journal
(
    idJournal        int not null,
    idUtilisateur     int not null,
    dateConnexion     datetime,
    dateDeconnexion   datetime,
    primary key (idJournal));

```



```

/*=====*/
/* Table: Message */
/*=====*/
create table Message
(
    idDiscussion    int not null,
    idMessage       int not null,
    dateMessage     datetime,
    contenu         varchar(254),
    statutMessage   bool,
    primary key (idDiscussion, idMessage)
);

/*=====*/
/* Table: Utilisateur */
/*=====*/
create table Utilisateur
(
    idUtilisateur   int not null,
    idProfil        int not null,
    login           varchar(35),
    password        varchar(35),
    statut          bool,
    acces           bool,
    nom             varchar(50),
    prenom          varchar(50),
    email           varchar(50),
    contact         varchar(15),
    primary key (idUtilisateur)
);

alter table Entreprise add constraint FK_adresseA foreign key (idFacture)
references Factures (idFacture) on delete restrict on update restrict;

alter table Entreprise add constraint FK_etreEffectueeSur foreign key (Pre_idFacture, idPrestation)
references Prestation (idFacture, idPrestation) on delete restrict on update restrict;

alter table Factures add constraint FK_concerner foreign key (idPayement)
references Payement (idPayement) on delete restrict on update restrict;

alter table Factures add constraint FK_generer foreign key (idUtilisateur, idAgent)
references Agents (idUtilisateur, idAgent) on delete restrict on update restrict;

alter table Journal add constraint FK_avoir foreign key (idUtilisateur)
references Utilisateur (idUtilisateur) on delete restrict on update restrict;

alter table Message add constraint FK_association1 foreign key (idDiscussion)
references Discussion (idDiscussion) on delete restrict on update restrict;

alter table Utilisateur add constraint FK_etreLieA foreign key (idProfil)
references Profil (idProfil) on delete restrict on update restrict;

alter table entretenir add constraint FK_entretenir foreign key (idDiscussion)
references Discussion (idDiscussion) on delete restrict on update restrict;

```

3. Quelques masques de saisie

- Ajout d'agent

Controller :

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Entreprises;
use App\Clients;
use App\User;
use App\Utilisateurs;
use App\Directeur;

class EntreprisesController extends Controller
{
    public function create()
    {
        return view('clients/Add_client');
    }

    public function enregistrer(Request $request){

        $entreprises = new Entreprises;
        $user = new User;
        $clients = new Clients;

        $entreprises->rs = $request->input('rs');
        $entreprises->adresse = $request->input('adresse');
        $entreprises->type = $request->input('type');
        $entreprises->save();
        @mkdir("dossiers");
        @mkdir("dossiers/clients");
        @mkdir("dossiers/clients/".$entreprises->rs);
        @mkdir("dossiers/clients/".$entreprises->rs."/factures");
        @mkdir("dossiers/clients/".$entreprises->rs."/documents");

        $user->email = $request->input('email');
        $user->password = bcrypt($request->input('password'));
        $user->name = $request->input('name');
        $user->prenom = $request->input('prenom');
        $user->numero = $request->input('numero');
        $user->poste = 'Directeur';
        $user->save();

        $clients->user_id = $user->id;
        $clients->entreprises_id = $entreprises->id;
        $clients->save();

        return redirect('/');
    }

    public function liste()
    {

```

```

    $entreprises=Entreprises::all();

    return view('entreprises/liste_entreprise')->with(['entreprises'=>$entreprises]);
}
}

```

Vue :

NOUVEAU AGENT

Renseignez correctement les informations

Renseignez les informations sur l'agent

Nom

Prénom

Email

nr (+226) 90 90 36 00

ENREGISTRER

Copyright © 2017 Par ADUAYOM MESSAN Ekoué Enoc
Tutoriel

Figure 39:formulaire d'ajout d'agent

Modèle :

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Agents extends Model
{
    protected $table = 'Agents';
    public $timestamps = true;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
}

```

```

protected $fillable = [
    'id','user_id',
];
/**
 * The attributes that should be hidden for arrays.
 *
 * @var array
 */
protected $hidden = [
    'password', 'remember_token',
];
}

```

- Discussion

Server.js :

```

var http = require('http');
var md5 = require('MD5');

httpServer= http.createServer(function(req,res){
    console.log('un utilisateur à affiché la page');

});
httpServer.listen(1334);

var io = require('socket.io').listen(httpServer);
var users={};
var messages=[];
var history=2;
io.sockets.on('connection',function(socket){

var me=false;
console.log('Nouveau utilisateur');
for(var k in users){
    socket.emit('newusr',users[k])
}

for(var k in messages){
    socket.emit('newmsg',messages[k])
}

/*
on a reçu un message
*/
socket.on('newmsg',function(message){
    message.user=me;
    date=new Date();
    message.h=date.getHours();
    message.m=date.getMinutes();
    messages.push(message);
    if (messages.length > history) {
        messages.shift();
    };
    io.sockets.emit('newmsg',message);

});
/*

```

```

je me connecte
*/
socket.on('login',function(user){
me = user;
me.id = user.username.replace('@','-').replace(':', '-');
me.avatar = 'https://gravatar.com/avatar/' + md5(user.username) + '?s=50';
socket.emit('logged');
users[me.id] = me;
io.sockets.emit('newusr',me);

});

/*
je quitte le chat
*/

socket.on('disconnect',function(){
if(!me){
return false;

}
delete users[me.id];
io.sockets.emit('disusr',me);
})
});

```

Clients.js :

```

(function($){

var socket = io.connect('http://localhost:1334');
var msgtpl = $('#msgtpl').html();
var lastmsg = false;
$('#msgtpl').remove();
$('#loginform').submit(function(event){
event.preventDefault();
socket.emit('login',{
username: $('#username').val(),
mail: $('#mail').val()
})
});

socket.on('logged',function(){
$('#login').fadeOut();
$('#message').focus();
});

/*
envois de message
*/
$('#form').submit(function(event){
event.preventDefault();
socket.emit('newmsg',{message: $('#message').val() });
});

```

```

    $('#message').val('');
    $('#message').focus();

})

socket.on('newmsg',function(message){
if(lastmsg!= message.user.id){
$('#messages').append('<div class="lv-item media right"></div>');
lastmsg = message.user.id;

}
$('#messages').append('<div class="lv-item media">' + Mustache.render(msgtpl,message) + '</div>');
$('#messages').animate({scrollTop:$('#messages').prop('scrollHeight')}, 500);
});

/*
gestion des connexions
*/
socket.on('newusr',function(user){
    $('#users').append('');

})

socket.on('disusr',function(){
$('#'+ user.id).remove();
})

})(jQuery);

```

Vue :

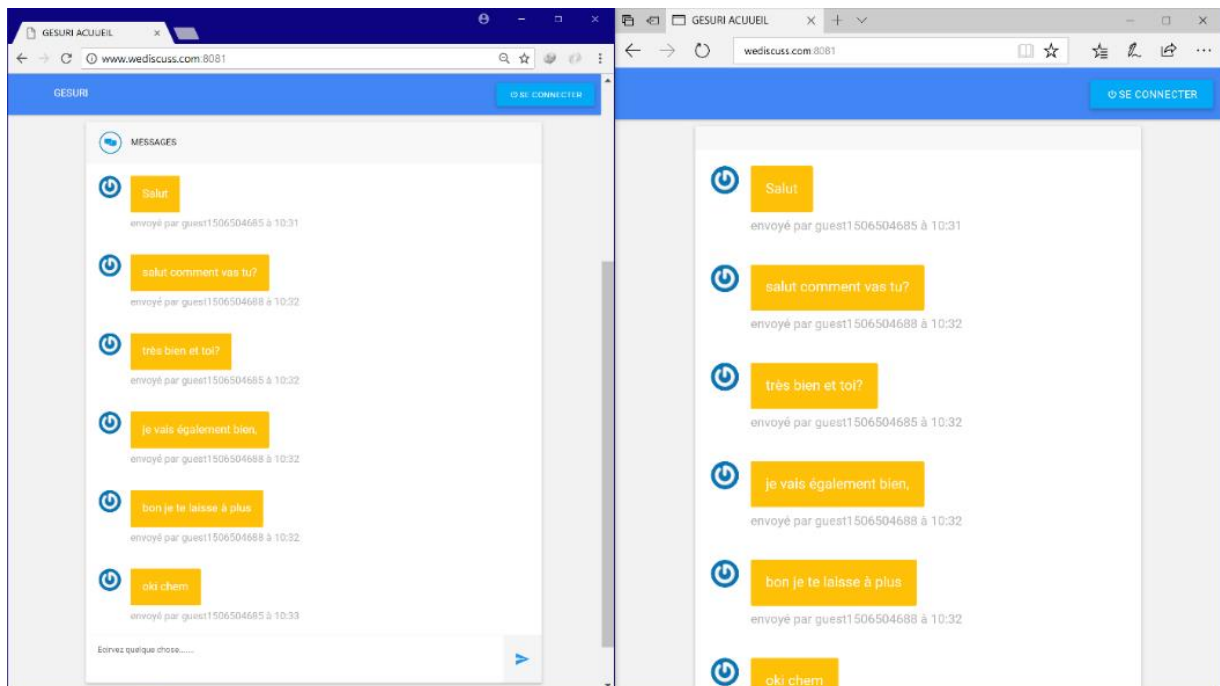


Figure 40:discussion instantanée

4. Quelques statistiques

- Aperçu globale du pourcentage de client par type d'entreprise

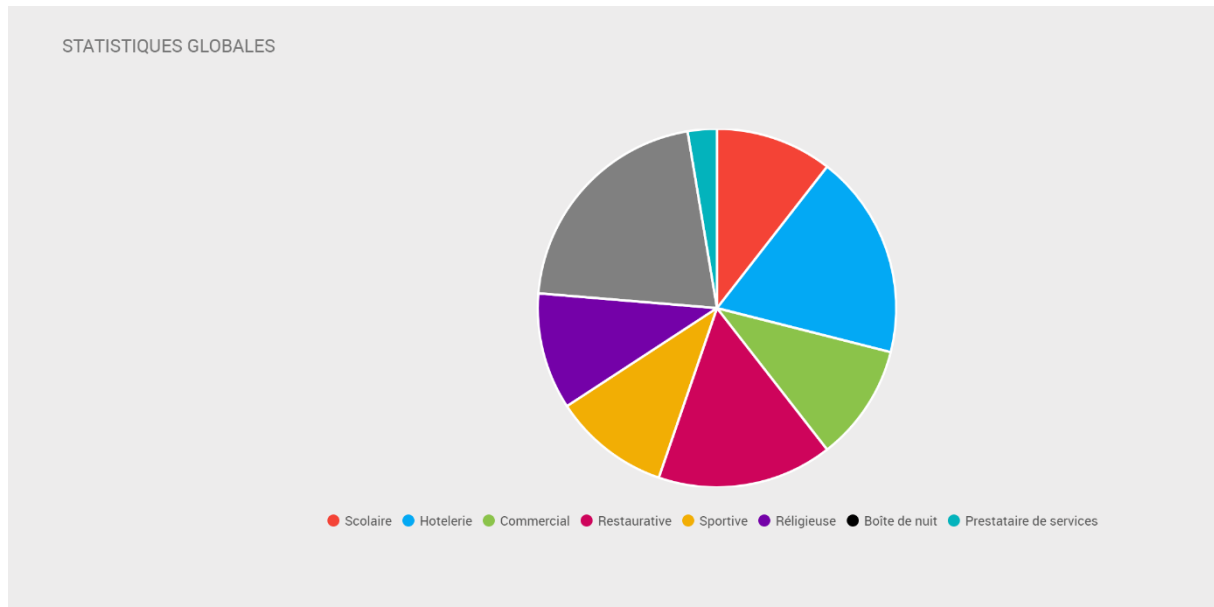


Figure 41: statistique aperçu global

- Aperçu individuel de pourcentage de client par type d'entreprise

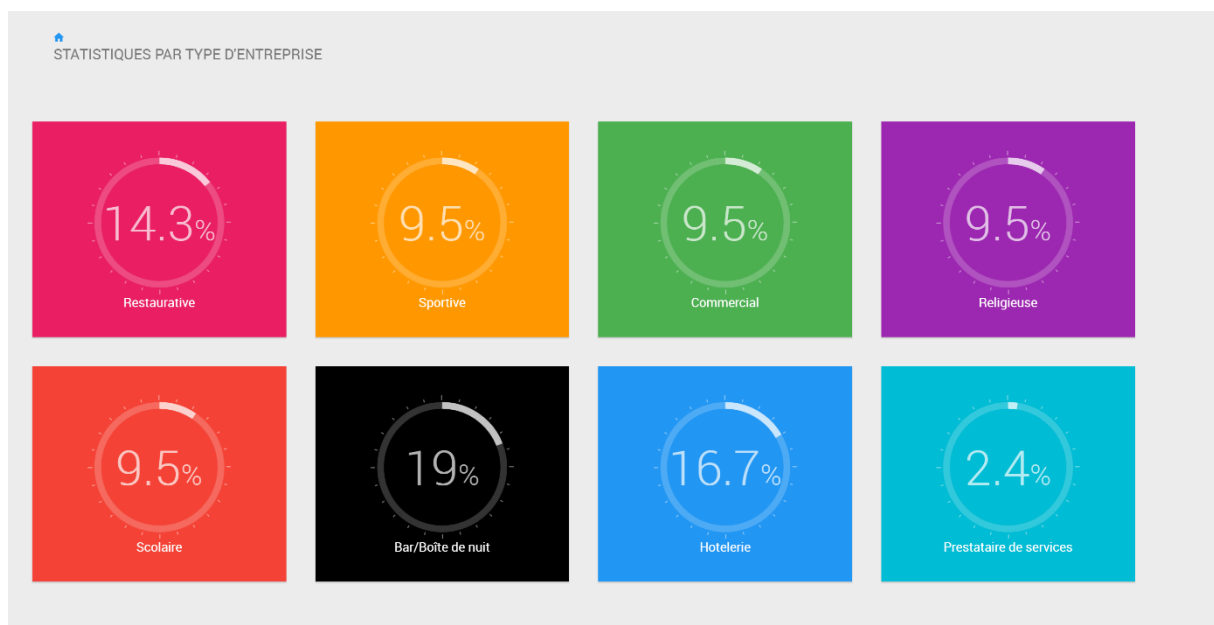


Figure 42: aperçu individuel

- Aperçu en bâton

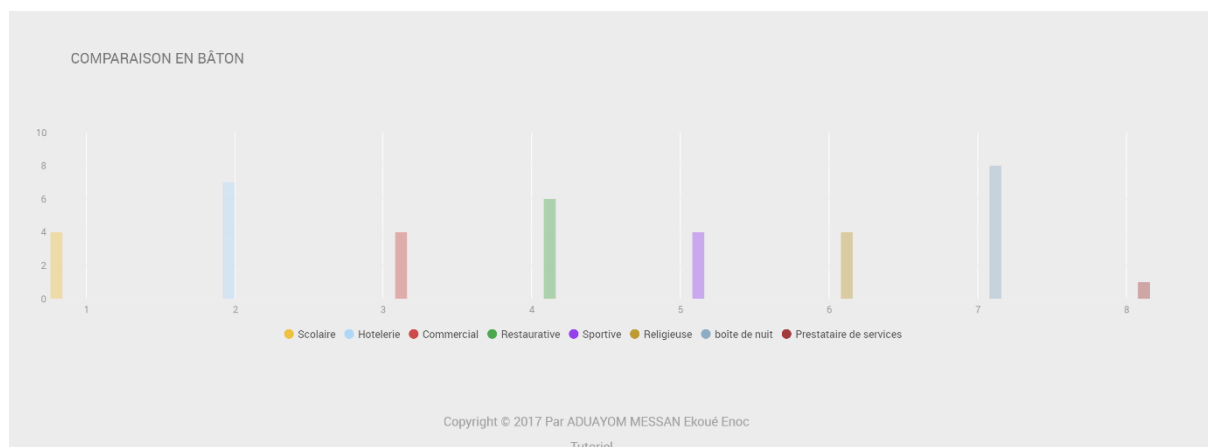


Figure 43:aperçu en bâton

III. PLANNING DE REALISATION DE L'APPLICATION

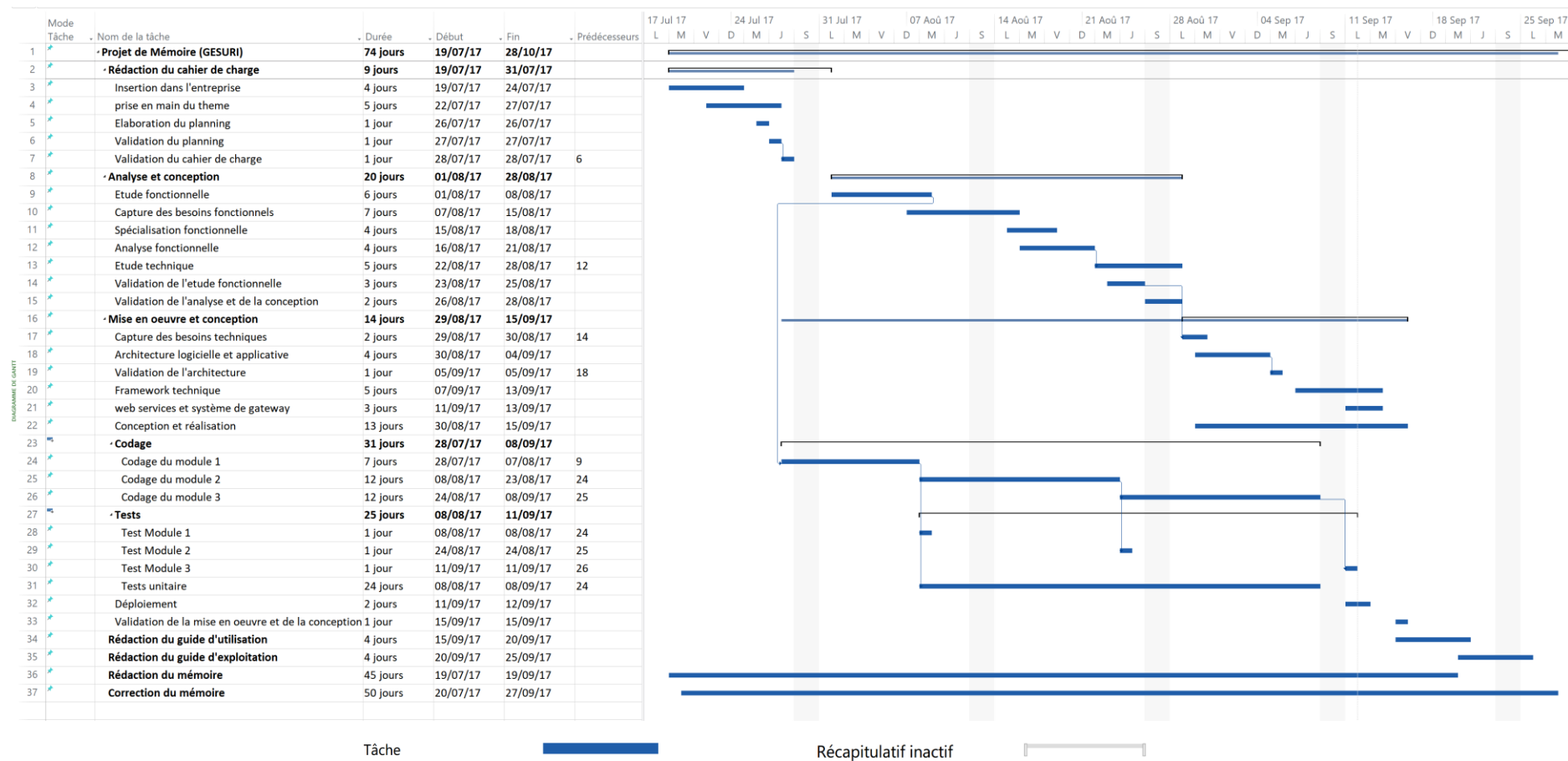


Figure 44:planning de réalisation

CONCLUSION

Ceci mets fin à la phase de mise en œuvre et réalisation de notre document. Dans cette partie, l'application a été sommairement présentée.

Nous allons maintenant passer à la quatrième partie de notre document soit, le guide d'exploitation.

PARTIE IV GUIDE D'EXPLOITATION

Du 15 Septembre 2017 Au 20 Septembre 2017

INTRODUCTION

Dans le dossier précédent, nous avons eu à présenter notre application et tous les outils qui nous ont permis de le mettre en place. Toutefois, il est nécessaire de remplir certaines conditions pour sa mise en production.

Dans ce dossier, nous allons, après avoir présenté la configuration matérielle et logicielle, présenter les normes de déploiement et de suivi.

I. CONFIGURATION LOGICIELLE ET MATÉRIELLE

1. Configurations matérielles

Les dispositions matérielles à prendre pour utiliser notre application est la suivante :

Tableau 15:configurations matérielles requise

	Configuration minimale	Configuration conseillé
Serveur	Serveur dédié Processeur Intel Xeon Core i5 2.5GHz, RAM 16Go, Disque 512Go SSD + 2*500Go HDD	Serveur dédié Processeur Intel Xeon Core i7 3.5GHz, RAM 32Go, Disque 512Go SSD + 4*500Go HDD
Client	Accès à internet	Accès à internet, navigateur Web récent

2. Configuration logicielle

Pour la mise en place de l'application, il est nécessaire d'installer les logiciels suivants :

- WampServer
- Composer
- Socket (Redis, Node.js, Mustache prérequis)

- **Installation de WampServer + MySQL**

Pour l'installation de WampServer il suffit d'aller sur le site www.wampserver.com de télécharger le package d'installation et de suivre les étapes d'installation. Faites attention à télécharger la version la plus stable. Il peut arriver qu'il y ait des erreurs de ports du fait que d'autres services utilisent déjà le port par défaut de WampServer soit le '8080' il peut être modifier très facilement. Les documentations peuvent être consultées sur le site en question. Il sera intégré à la suite de l'installation une version de MySQL.

- **Installation de composer**

LARAVEL nécessite Composer pour pouvoir fonctionner. Pour l'installation de celui-ci il faut se rendre sur le site www.getcomposer.org , télécharger le package d'installation et l'exécuter tout en étant connecté à internet.

- **Socket**

Pour implémenter la discussion instantanée, Socket devient nécessaire. Avant son installation, d'autres logiciels sont nécessaires.

- Pour l'installation de Node.js, il faut se rendre sur le site www.Nodejs.org et télécharger la version la plus stable et suivre les étapes d'installation.
- Pour l'installation de Redis, vous pouvez vous rendre sur www.redis.io pour avoir toutes les version sinon, pour avoir un version stable se rendre sur <https://github.com/MicrosoftArchive/redis> pour télécharger un fichier compressé ou pour télécharger une version en package d'extension MSI se rendre sur <https://github.com/MicrosoftArchive/redis/releases>. Pendant l'installation n'oubliez pas de cocher la case comme indiqué ci-dessous.

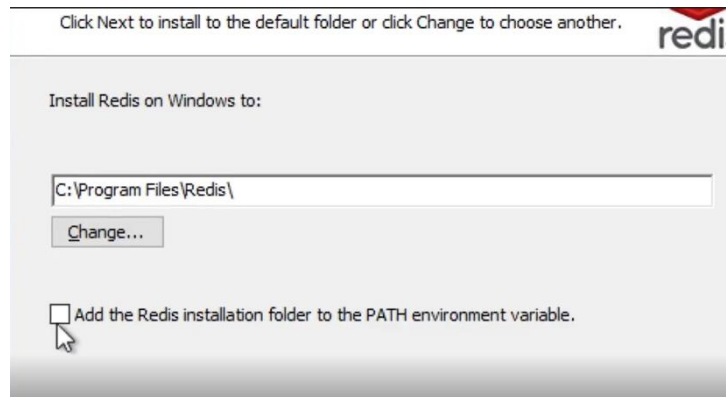


Figure 45: installation Redis

- Pour installer Socket, lancer le server Node.js dans la palette d'application de Windows, et exécutez la commande « `npm install express redis socket.io --save` ».

II. DEPLOIEMENT ET SUIVI

1. Déploiement de l'application

Pour accéder à l'application web, il faudra entrer une URL dans la barre d'adresse d'un navigateur web.

- Nous allons donc commencer par acheter le nom de domaine qui permettra d'accéder à l'application,
- Ensuite nous devons paramétrer les entrées DNS afin que le serveur puisse gérer les redirections,
- Maintenant que nous disposons du nom de domaine, nous allons reprendre les étapes de déploiement du serveur web local à savoir :
 - Installer Composer, le gestionnaire de composants de Laravel,
 - Cloner le dossier contenant l'application et donner les droits d'accès,
 - Exécuter la commande « composer update » pour télécharger les librairies dont a besoin l'application qui seront mises dans le dossier vendor,
 - Créer la base de données sur laquelle va tourner l'application, créer un utilisateur et lui donner tous les droits sur la base,
 - Importer les tables vers la base de données que l'on a créée,
 - Éditer le fichier de configuration du projet Laravel d'extension env (. Env),
 - Éditer le fichier de configuration du site : WeDiscuss.conf ;

2. Suivi de l'application

- Sauvegarde de l'application

Pour pouvoir subvenir ou réagir efficacement en face de certains problèmes, il est donc nécessaire d'instaurer une politique de sauvegarde. Il existe deux (2) types de sauvegarde :

- La sauvegarde logique
 - La sauvegarde physique
- 1- Sélectionner la base de données **WeDiscuss** en cliquant sur son nom dans la colonne de gauche.
 - 2- Dans les onglets en haut de l'application, sélectionnez "Exporter"
 - 3- Dans le champ "Exporter" de la page, assurez-vous que SQL soit sélectionné puis cliquez sur "Tout sélectionner". Vous sélectionnez en fait les tables de votre base de données. Puisque vous recherchez à faire une sauvegarde complète de votre base, il convient donc de tout sélectionner mais il est possible d'en sélectionner une seule à chaque fois et de répéter l'opération.
 - 4- Cochez "Structure". Il s'agit là de sauvegarder le nom des tables, et le nom des champs de la table c'est à dire comment est arrangée votre base de données. En quelque sorte, cela représente un tableau destiné à recevoir des données mais sans le contenu. Cocher les cases « inclure la valeur constante de l'AUTO_INCREMENT » et « protéger les noms des tables et des champs par des "" ».
 - 5- Cochez "Données". Cette fois il s'agit du contenu de votre base, le plus important en quelque sorte. Les options "insertions complètes" et "insertions étendues" ne sont pas cochées. Rassurez-vous, que vous ayez coché ou non ces options en changera en rien le contenu sauvegardé, c'est seulement la façon dont il sera sauvegardé qui changera.

6- Cochez "Transmettre". Important, car si vous ne cochez pas cette case, PhpMyAdmin ne vous proposera pas de télécharger la sauvegarde mais se contentera de vous l'afficher. Bon à savoir, mais ce n'est pas notre but ici. Dans le cas d'une sauvegarde de bases de données importantes, vous avez la possibilité de "zipper" (.zip) ou de "gzipper" (.gz) votre fichier de sauvegarde (par défaut au format .SQL), c'est à dire de le compresser pour obtenir un fichier moins lourd.

7- Enfin, cliquez sur "Exécuter" et patientez pendant la préparation de votre sauvegarde jusqu'à ce qu'une fenêtre de téléchargement vous propose de récupérer le fichier généré. Sauvegardez-le fichier .SQL obtenu ou .zip ou .gzip.

En effet, après la sauvegarde il faut penser à une restauration ultérieure. Cette partie sera abordée dans le guide d'utilisation.

- **Suivi des évènements**

Pour permettre à l'administrateur de pouvoir suivre le bon déroulement des traitements, il est prévu un document dans lequel les pannes qui subviennent au cours de l'exploitation seront mentionnées. L'administrateur ne pourra pas tout constater.

Voici un exemple de la fiche :

Tableau 16:fiche de renseignement d'erreurs

[illegible]

CONCLUSION

Dans ce document, ont été décrites, les actions de suivi de l'application soit les sauvegardes et le suivi des pannes, de répertorier les différentes failles de l'application afin d'effectuer des maintenances, les configurations pouvant permettre l'exploitation de l'application.

PARTIE V

GUIDE D'UTILISATION

Du 20 Septembre 2017 Au 25 Septembre 2017

INTRODUCTION

Cette partie sert de référence à l'application. Il s'agit ici d'une documentation de l'application pour expliquer son utilisation. Il est nécessaire pour la prise en main du système et permet de se familiariser avec celle-ci.

I. PRÉSENTATION DE L'APPLICATION

1. Lancement de l'application

Au lancement de l'application nous avons une page, cette page permet aux futurs clients de pouvoir discuter sur la plateforme. Il s'agit d'une forme de forum instantanée en invité.

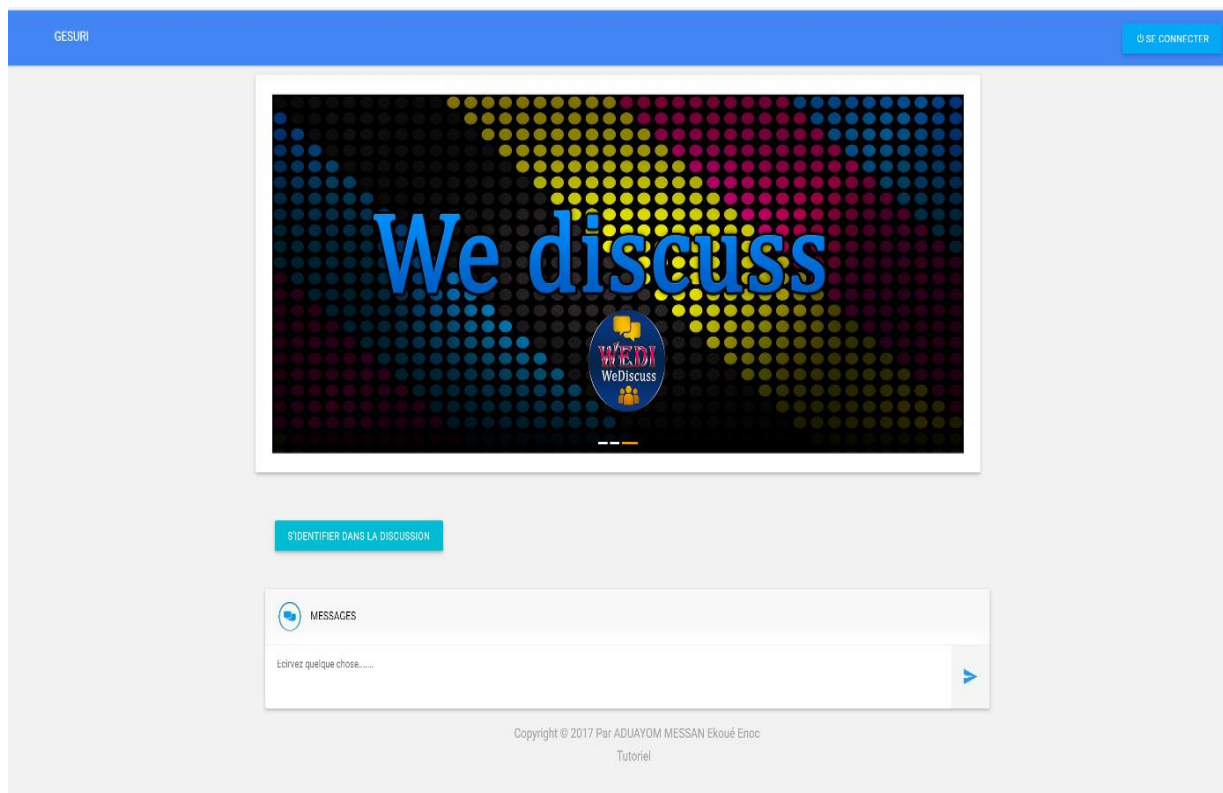


Figure 47:page d'accueil

2. Connexion

Le client ou l'agent désirant se connecter peut grâce au bouton « se connecter » situé dans le coin haut à droite de la page lancer la page de connexion.

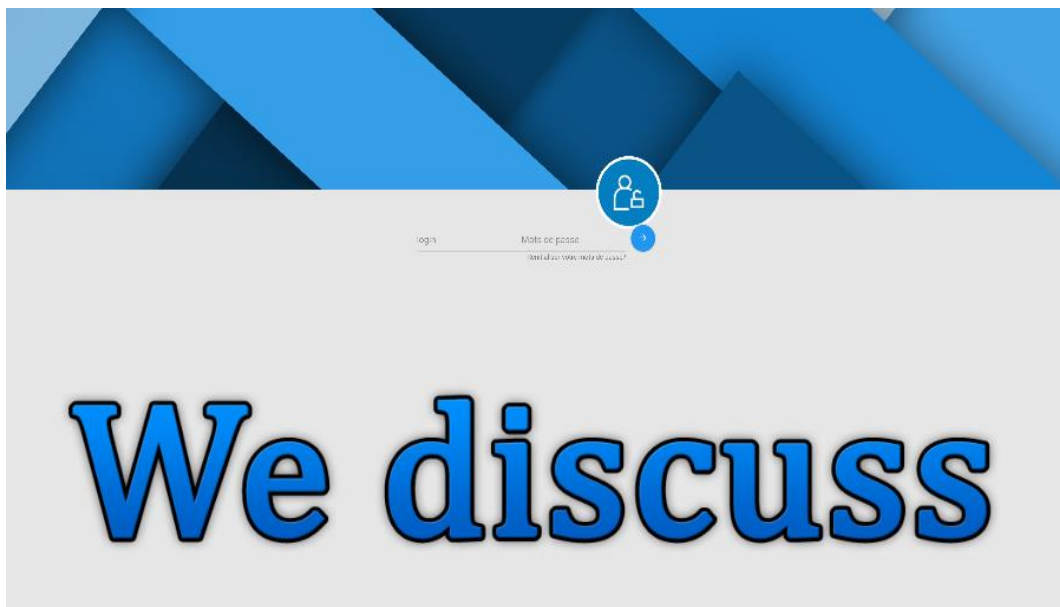


Figure 48:page de connexion

3. Présentation des menus

Selon les droits d'accès de chacun, il a accès à un menu bien spécifique.

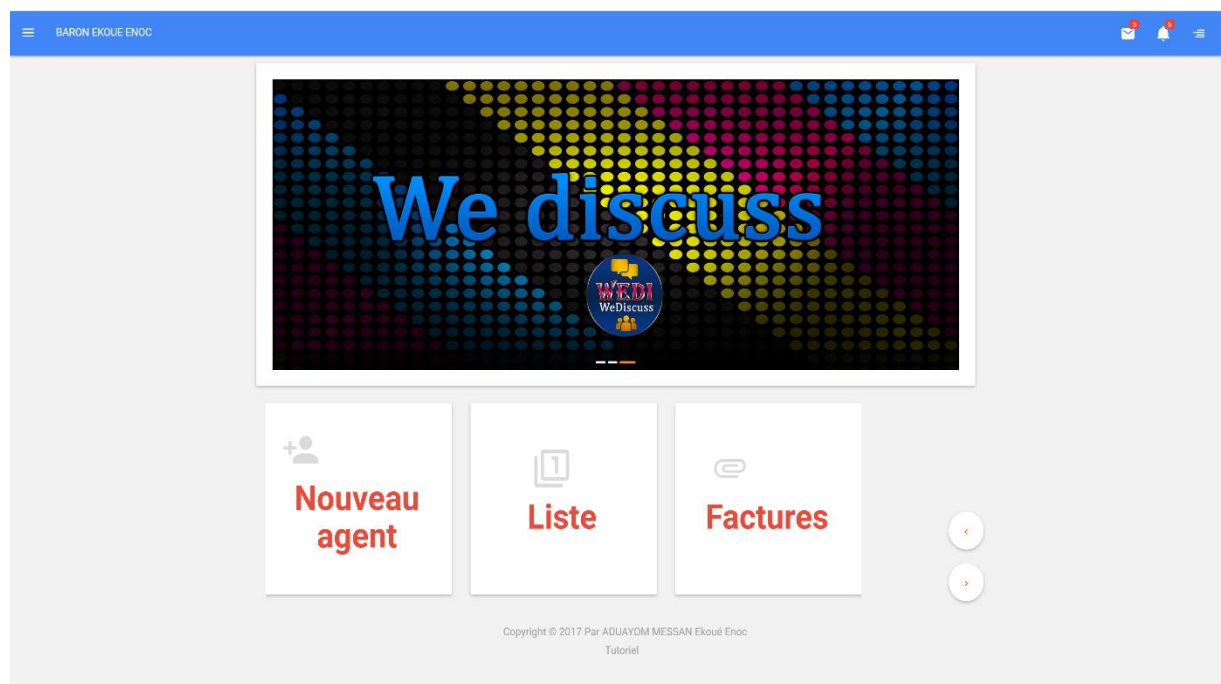


Figure 49:menu de l'administrateur

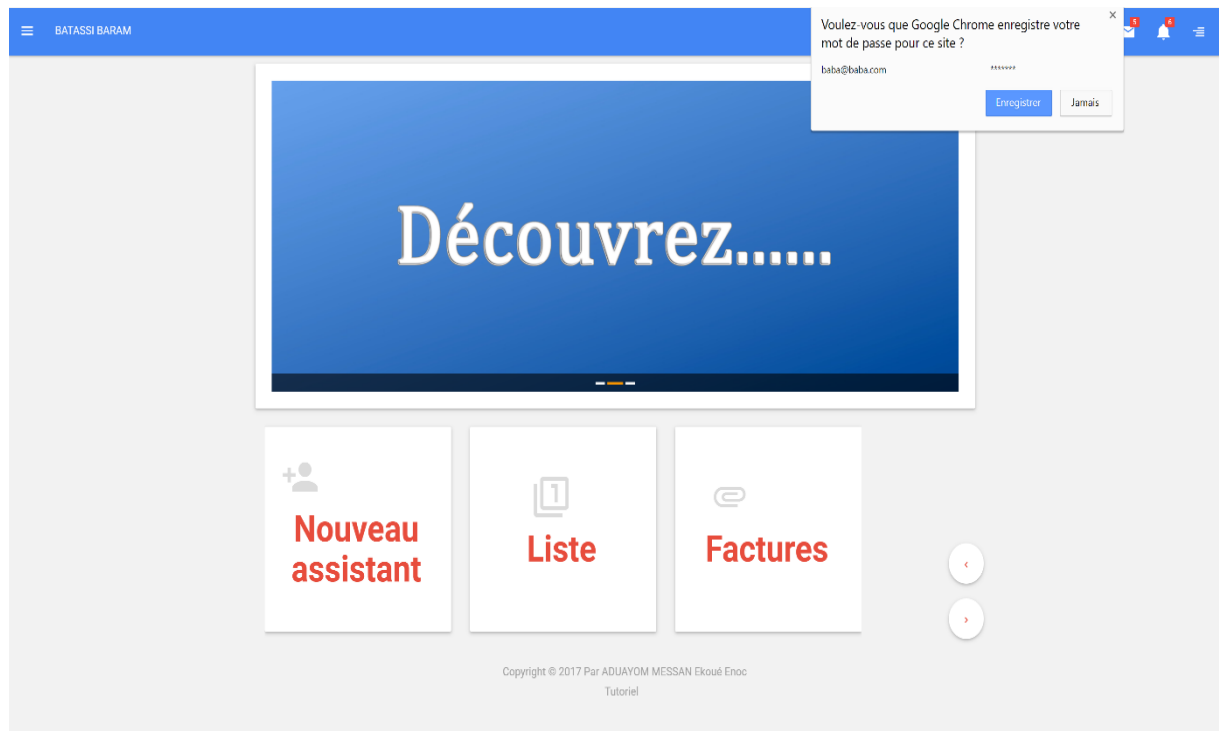


Figure 50:menu d'un agent

L'application par ses menus très identificateurs décrit exactement l'action qui sera mener derrière chaque bouton soit :

- Ajouter un client qui renvoie vers l'ajout d'un client,
- Factures qui renvoie vers une galerie d'image et de fichiers.


Par ailleurs pour pouvoir faciliter la navigation sur chaque page de l'application, un menu est également disponible d'un léger clic sur le bouton haut coin gauche de l'image ci-dessus. Au clic on a le menu suivant :



Figure 51:menu modal

Il est également possible d'avoir accès à ses ajouts puis aux informations puis aux informations modifiables. Selon le droit d'accès de chaque utilisateur nous avons les fenêtres suivantes :

BARON EKOUÉ ENOC
Admin


1562
Total Connexions


Contacts
+4567345985
baron@baron.fr

INFORMATIONS		MODIFIER
Informations basiques		
Nom	BARON	
Prenom	Ekoue Enoc	
Poste	Admin	
société	GESURI	
Login	baron@baron.fr	

Copyright © 2017 Par ADUAYOM MESSAN Ekoué Enoc
Tutoriel

Figure 52: info client

BATASSI BARAM
Directeur

AUCUNE IMAGE DISPONIBLE

1562
Total Connexions

Contacts
+22893245632
baba@baba.com

INFORMATIONS		MODIFIER	MES AJOUTS
Informations basiques			
Nom	BATASSI		
Prenom	Baram		
Poste	Directeur		
société	DIGITCORP		
Login	baba@baba.com		

Copyright © 2017 Par ADUAYOM MESSAN Ekoué Enoc
Tutoriel

Figure 53: info agents

II. MAINTENANCES

Les erreurs que peuvent rencontrer notre application sont les suivantes :

1. Les erreurs courantes

Comme il a été annoncé dans le document précédent, il peut avoir de erreurs au niveau de la base de données, il faut alors penser à faire une restauration. D'ailleurs une sauvegarde journalière serait conseillée.

A part cette erreur, les erreurs qui peuvent être encore rencontrées sont les suivantes :

Tableau 17:recueil d'erreurs

Description	Action à mener
Vous n'êtes pas autorisé sur cette page	Vérifier si vous êtes connecté, sinon connectez-vous ou cliquez sur le bouton home
Page inaccessible	Vérifiez si le lien saisie est correct.
Pas de réponse au clic d'un navigateur	Vérifiez si votre navigateur est à jours

2. Autres erreurs

En cas d'autres erreurs renseignez les sur la fiche de renseignement d'image (figure 16).

CONCLUSION

Une solution informatique doit pouvoir se rapprocher le plus possible des désirs du client et répondre à ses besoins dans le plus optimal des cas possibles. Pour avoir la main sur une application, il faut tout d'abord s'y habituer. Le guide d'utilisation est le premier pas vers cette familiarisation avec l'application.

CONCLUSION GENERALE

Ce document est le produit du projet informatique qui nous a été confié pendant notre stage de trois mois à GESURI SARL. Il est divisé en cinq grandes phases dont les phases d'étude, de conception et de réalisation qui représentent le pivot du processus de développement logiciel. En se référant aux objectifs visés, nous avons développé une application capable de répondre efficacement aux attentes du cabinet. Cette application est développée grâce au langage PHP, à Laravel et aux technologies Node.js et socket. L'application développée pendant ce stage servira à gérer les clients et les discussions entre ceux-ci et le cabinet. Elle permettra au cabinet de réduire en coût et en temps de traitement certaines tâches, de garantir la sécurité, la fiabilité, la disponibilité ainsi que la réutilisabilité des données.

Ce stage nous a fait découvrir de nouvelles technologies et nous a permis d'acquérir de nouvelles connaissances.

- Bibliographie

• Ouvrages :

- UML 2 en action De l'analyse des besoins à la conception : Pascal Roques ;

• Notes de cours :

- Analyse (UML), M. GBINU (Année académique 2015-2016) ;
- Programmation Web, M. Mélouziba DIZEWE (Année académique 2015-2016);
- Introduction au Génie Logiciel : M. Mélouziba DIZEWE (2015-2016).

• Sites Web consultés

SITES	DATES ET HEURES
https://redis.io/	19 Septembre à 17h00
https://socket.io/	19 Septembre à 17h30
https://stackoverflow.com/questions/38298502/laravel-5-2-live-chat-with-socket-io-and-redis-private-message	22 Septembre à 9h00
https://nodejs.org/en/download/	21 Septembre à 22h00
https://laravel.com/Query%20Builder.html	10 Septembre à 15h
https://www.grafikart.fr/forum/topics/8372	22 Septembre à 10h00
http://www.nowsms.com	05 Septembre à 11h00

• Anciens documents consultés

Gestion informatisée des affaires et des informations d'un cabinet : cas de C.I.D.O (TOGBETSE Israël 2015-2016),

Gestion automatisée des inscriptions et des Études du centre CIFOP (Lawson-Body Jonathan & Napporn Omer 2013-2014).

TABLE DES MATIERES

DEDICACES.....	I
REMERCIEMENTS	II
AVANT-PROPOS	III
RESUME	IV
ABSTRACT	V
SOMMAIRE.....	VI
INTRODUCTION GENERALE.....	VIII
LISTE DES PARTICIPANTS	IX
LISTE DES TABLEAUX	X
LISTE DES FIGURES	XI
GLOSSAIRE	XIII
PARTIE I : CAHIER DE CHARGE.....	1
INTRODUCTION	2
I. PRESENTATIONS	3
1. Présentation de l'IAI-TOGO.....	3
2. Présentation de GESURI Sarl	4
II. THEME DE STAGE	7
1. Présentation du sujet	7
2. La problématique du sujet	7
3. Intérêt du sujet	8
III. ETUDE DE L'EXISTANT	9

IV. CRITIQUE DE L'EXISTANT	10
V. PROPOSITION DES SOLUTIONS.....	11
1. Évaluations techniques	11
2. Évaluations financières	15
VI. CHOIX DE LA SOLUTION	16
VIII. PLANNING PREVISIONNEL DE REALISATION	17
CONCLUSION	18
PARTIE II : ANALYSE ET CONCEPTION	19
INTRODUCTION	20
I. PRESENTATION DES METHODES D'ANALYSE.....	21
1. Langage UML.....	21
2. Processus 2TUP	22
II. PRESENTATION DE L'OUTILS DE MODELISATION	24
1. Qu'est-ce que Power Designer.....	24
2. Modélisation avec Power Designer.....	25
III. ETUDE DETAILLE DE LA SOLUTION.....	26
1. Diagrammes de cas d'utilisation	26
2. Diagrammes de séquence	45
3. Diagrammes d'activités.....	49
4. Diagrammes de classe.....	53
CONCLUSION	56
PARTIE III : MISE EN ŒUVRE ET REALISATION	57
INTRODUCTION	58
I. MISE EN ŒUVRE	59
1. Matériels et logiciels utilisés	59
2. Architecture matérielle et logicielle de l'application	68
3. Déploiement de l'application.....	70
4. Sécurité de l'application.....	71
II. PRÉSENTATION DE L'APPLICATION.....	72
1. Présentation.....	72

2. Script de création de la base de données	74
3. Quelques masques de saisie - Ajout d'agent	76
4. Quelques statistiques.....	81
III. PLANNING DE REALISATION DE L'APPLICATION	83
CONCLUSION	84
PARTIE IV : GUIDE D'EXPLOITATION	85
INTRODUCTION	86
I. CONFIGURATION LOGICIELLE ET MATERIELLE	87
1. Configurations matérielles	87
2. Configuration logicielle	87
II. DEPLOIEMENT ET SUIVI	90
1. Déploiement de l'application	90
2. Suivi de l'application	91
CONCLUSION	94
PARTIE IV : GUIDE D'UTILISATION.....	95
INTRODUCTION	96
I. PRESENTATION DE L'APPLICATION	97
1. Lancement de l'application	97
2. Connexion	97
3. Présentation des menus	98
II. MAINTENANCES.....	102
1. Les erreurs courantes.....	102
2. Autres erreurs	102
CONCLUSION	103
CONCLUSION GENERALE	104
TABLE DES MATIERES.....	II