

REPUBLIQUE DU SENEGAL

Un Peuple – Un But – Une Foi



UNIVERSITE CHEIKH ANTA DIOP DE DAKAR



ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT GENIE INFORMATIQUE

MEMOIRE DE FIN DE CYCLE

Pour l'obtention du :

DIPLOME D'INGENIEUR DE CONCEPTION (D.I.C) EN INFORMATIQUE

SUJET :

Création d'un ERP orienté CRM pour la gestion de la relation client

Lieu de stage :



Période stage : 15/03/2021 – 15/07/2021

*Présenté et soutenu par
Anne Marie DIOP*

*Professeur encadrant
M. Alassane BAH*

*Maître de stage
Mamadou THIANDOUM*

REPUBLIQUE DU SENEGAL

Un Peuple – Un But – Une Foi



UNIVERSITE CHEIKH ANTA DIOP DE DAKAR



ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT GENIE INFORMATIQUE

MEMOIRE DE FIN DE CYCLE

Pour l'obtention du :

DIPLOME D'INGENIEUR DE CONCEPTION (D.I.C) EN INFORMATIQUE

SUJET :

Création d'un ERP orienté CRM pour la gestion de la relation client

Lieu de stage :



Période stage : 15/03/2021 – 15/07/2021

*Présenté et soutenu par
Anne Marie DIOP*

*Professeur encadrant
M. Alassane BAH*

*Maître de stage
Mamadou THIANDOUM*

DEDICACES

Je dédie ce mémoire :

- ✓ *A **Eliane DIOUF**, une femme brave, généreuse et super attentionnée. Maman, tu es tellement gentille et ouverte d'esprit qu'on te prend parfois pour une copine. Nous ne pouvions espérer mieux comme mère. Je remercie le seigneur pour ta vie.*
- ✓ *A **Jean Pierre DIOP**, un homme d'une intelligence et d'une humilité extraordinaire. Tu as tout fait pour qu'on ne manque de rien.*
- ✓ *A ma Tante **Sr Marie DIOUF**, une seconde maman qui a toujours été là par ses conseils, ses prières, son soutien sur tous les plans.*
- ✓ *A ma grande sœur **Agathe Marame**, ma meilleure amie, mon modèle, je t'aime beaucoup.*
- ✓ *A ma grande sœur **Caroline**, mon sosie, je t'adore.*
- ✓ *A **Irène Gazal PEREIRA**, une sœur d'une autre mère.*
- ✓ *A **Éric Auxence MBENGUE**, un don du ciel, une source de joie.*
- ✓ *A **Pape Idrissa NDOYE**, un ami juste formidable.*
- ✓ *A Toute mes amies, **Peinda FALL, Fatou BA SOW, Saly SARR, Ndeye Maty DABO**, la liste est longue. Les moments que je passe avec vous sont toujours magnifiques. Vous êtes les meilleures, je vous aime.*
- ✓ *A toute ma famille : mes tantes, mes oncles, mes cousins et cousines,*
- ✓ *A tous les étudiants de la promo DIC 2018-2021.*

REMERCIEMENTS

A la fin de cette formation je tiens à remercier tous ceux qui ont de près ou de loin contribué à la réalisation de ce travail et particulièrement :

- ✓ Ma famille et mes amis pour leur soutien et encouragement tout au long de mes études.
- ✓ Tout le personnel de SOPARTECH pour leur confiance en moi, leurs encouragements, leur aide, pour la motivation qu'ils m'apportent et pour m'avoir appris beaucoup de choses.
- ✓ J'exprime ma profonde gratitude à M. Mamadou THIANDOUM et M. Issa BALDE pour leur disponibilité, et pour l'expérience acquise. Merci beaucoup à vous pour tout ce que vous avez fait.
- ✓ M. Alassane BAH pour son encadrement et toute son aide, merci beaucoup.
- ✓ Mes remerciements vont également à l'ensemble de l'équipe pédagogique du département génie informatique de l'Ecole Supérieure Polytechnique de Dakar pour son accompagnement et l'enseignement dispensé.

SOMMAIRE

AVANT-PROPOS.....	VIII
RESUME.....	ERREUR ! SIGNET NON DEFINI.
ABSTRACT.....	ERREUR ! SIGNET NON DEFINI.
INTRODUCTION	1
CHAPITRE I : PRESENTATION GÉNÉRALE	2
I. PRESENTATION DE LA STRUCTURE D'ACCUEIL [1].....	2
A. SOCIETE.....	2
B. STRATEGIE	2
C. SOLUTIONS.....	3
D. PRESENTATION DE MIKADO FRAMEWORK :	3
II. PRESENTATION DU SUJET	4
A. CONTEXTE.....	4
B. PROBLEMATIQUE.....	4
C. OBJECTIFS	5
i. Objectif global.....	5
ii. Objectifs spécifiques	5
D. ETAT DE L'ART DU CRM ET DE L'ERP	5
i. Définition	5
ii. Fonctionnalités de base d'un CRM.....	6
III. METHODE DE TRAVAIL	6
A. DEFINITION DE CONCEPTS.....	6
i. L'analyse	6
ii. La conception.....	7
B. CLASSIFICATION DES METHODOLOGIES DE GESTION DE PROJET [3]	7
C. CHOIX D'UNE METHODOLOGIE DE GESTION DE PROJET	8
i. Présentation de Scrum [4].....	8
ii. Application de Scrum à notre projet	9
iii. Présentation du langage de modélisation UML [5]	10
CHAPITRE 2 : SPECIFICATION ET ANALYSE DES BESOINS	12
I. ETUDE DE L'EXISTANT	12
A. PRESENTATION DE L'APPLICATION	12
B. ETUDE CRITIQUE DE L'EXISTANT	16
II. ACTEURS ET ROLES	17
III. LES BESOINS FONCTIONNELS	17
IV. LES BESOINS NON FONCTIONNELS.....	18
V. MODELISATION	18
A. PRESENTATION DES FONCTIONNALITES.....	19
B. DESCRIPTION DES FONCTIONNALITES	22
C. DIAGRAMME DE CLASSE	26
VI. GESTION DE PROJET	29
CHAPITRE 3 : CONCEPTION DE LA SOLUTION	32

I.	OUTILS ET TECHNOLOGIES	32
A.	J2EE [6]	32
B.	LE LANGAGE JAVA	33
C.	PRESENTATION DE QUELQUES COMPOSANTS	33
i.	JSP.....	33
ii.	Servlets [7].....	33
e.	Bootstrap [7].....	34
f.	jQuery [8].....	35
II.	ARCHITECTURE DE LA SOLUTION	35
CHAPITRE 4 : PRESENTATION DE LA SOLUTION.....		37
I.	PAGE D’ACCUEIL	37
II.	ESPACE DE PARAMETRAGE	37
A.	GESTION DES PROFILS	38
B.	GESTION DES UTILISATEURS	38
C.	GESTION DES TABLES PHYSIQUES.....	39
D.	CONFIGURATION DU SERVEUR SMTP	40
II.	GESTION DES CLIENTS.....	41
A.	AJOUT D’UN CLIENT	41
B.	AFFICHAGE POST IT D’UN CLIENT SPECIFIQUE GRACE AU FILTRE.....	42
C.	AFFICHAGE SOUS FORME D’ONGLETS D’UN CLIENT SPECIFIQUE GRACE AU FILTRE	43
D.	GESTION DE L’AGENDA	43
E.	ENVOI DE MAILS	44
CONCLUSION		46
REFERENCES		47
RESUME.....		48
ABSTRACT.....		49

SIGLES ET ABRÉVIATIONS

BDD	Base de Données
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
J2EE	Java 2 Enterprise Edition
JSP	Java Server Page
NIC	Numéro Interne de Classement
MVC	Modèle Vue Contrôleur
PPE	Personne Politiquement Exposée
PTIA	Perte Totale et Irréversible d'Autonomie
SGBD	Système de Gestion de Base de Données
SIRET	Système d'Identification du Répertoire des Etablissements
SMTP	Simple Mail Transfer Protocol
RAD	Rapid Application Development
SDK	Software Development Kit
SQL	Structured Query Language
UML	Unified Modeling Language

TABLE DES FIGURES

Figure 1: Les treize diagrammes d'UML	11
Figure 2 : Page d'accueil de l'application Alias CRM	13
Figure 3 : Liste des clients	13
Figure 4 : Ajout d'un client de l'application Alias CRM.....	14
Figure 5 : Liste des registres de commerce des identités morales de Alias CRM	15
Figure 6 : Envoi d'emails dans l'application Alias CRM	16
Figure 7 : Diagramme de package de notre application	19
Figure 8 : Diagramme de cas d'utilisation du paramétrage de l'application	20
Figure 9 : Diagramme de cas d'utilisation du paramétrage de données	20
Figure 10 : Diagramme de cas d'utilisation de la gestion des clients	21
Figure 11 : Diagramme de cas d'utilisation de la gestion des fiches d'un client	21
Figure 12 : Diagramme de séquence de l'enregistrement d'un client	22
Figure 13 : Fiche textuelle de l'enregistrement d'un client	23
Figure 14 : Diagramme de séquence de la modification d'une fiche client	24
Figure 15 : Fiche textuelle de la modification d'une fiche client	24
Figure 16 : Diagramme de séquence de la gestion du dossier fonctionnel	25
Figure 17 : fiche textuelle de la gestion du dossier fonctionnel	26
Figure 18: Diagramme de classe des informations personnelles du client	27
Figure 19 : : Diagramme de classe des informations sur les biens et la sante du client	28
Figure 20 : Diagramme de classe du paramétrage de l'application	28
Figure 21 : Architecture de notre application	Erreur ! Signet non défini.
Figure 22 : Page d'accueil	37
Figure 23 : Paramétrage des profils	38
Figure 24 : Ajout d'un utilisateur	39
Figure 25 : Ajout d'une nouvelle table	39
Figure 26 : Paramétrage des champs d'une table	40
Figure 27 : Configuration du serveur SMTP	41
Figure 28 : Ajout d'un client	42
Figure 29 : Filtre pour les clients	42
Figure 30 : Consultation des informations d'un client	43
Figure 31 : Affichage des informations d'un client sous forme d'onglets.....	43
Figure 32 : Gestion de l'agenda.....	44
Figure 33 : Envoi de mails	44

LISTE DES TABLEAUX

Tableau 1 : : Backlog Produit de notre application Sprint 1	29
Tableau 2 : : Backlog Produit de notre application Sprint 2	30
Tableau 3 : : Backlog Produit de notre application Sprint 3	30

AVANT-PROPOS

L'Ecole Supérieure Polytechnique de Dakar (ESP) est un établissement public de formation professionnelle dotée d'une personnalité juridique et d'une autonomie financière. Elle fait partie intégrante de l'Université Cheikh Anta Diop de Dakar (UCAD). Elle a été créée en 1994 et a pour mission de former tant sur le plan théorique que pratique des techniciens supérieurs, des ingénieurs technologues, des ingénieurs de conception, des managers en gestion d'entreprise et des docteurs ; de dispenser un enseignement supérieur en vue de préparer aux fonctions d'encadrement dans divers domaines telle la production, la recherche appliquée...

L'ESP compte (6) départements dont le département informatique où, pour l'obtention du Diplôme d'Ingénieur de Conception (DIC) à l'issue de trois (3) années, les étudiants doivent effectuer un stage de fin de formation dans une entreprise.

Ce stage permet à l'étudiant de renforcer son savoir et surtout d'acquérir un savoir-faire, tout en essayant d'adapter ses connaissances aux cadres de la vie professionnelle. Il lui permet d'avoir une vision pratique de la formation reçue et donne lieu à la rédaction d'un mémoire de stage présentant les tâches accomplies.

C'est à l'issue de ce stage effectué à SOPARTECH que nous avons eu à rédiger ce présent mémoire. Notre stage porte sur la création d'un ERP orienté CRM pour la gestion de la relation client.

Introduction

Actuellement, l'environnement des entreprises et multinationales est tel que, pour rester compétitives et efficaces, il faut constamment trouver des stratégies qui s'adaptent aux réalités du marché. Les entreprises doivent également faire face à beaucoup d'enjeux tels que l'optimisation de la qualité de leurs produits, l'amélioration des relations avec leurs clients, etc.

C'est dans cette optique que SOPartech Développement a voulu mettre en place une application CRM (Customer Relationship Management) qui facilite la gestion de la relation client aux entreprises.

En effet, un CRM est un outil qui centralise les informations des clients et permet d'avoir une vue d'ensemble sur les relations et interactions avec eux. Il peut aussi être vu comme une philosophie d'entreprise qui prône la centralisation de son activité autour du client dans le but d'être plus productif, performant et d'améliorer les relations commerciales au quotidien.

La plupart des entreprises utilisent un CRM mais l'enjeu demeure d'en trouver un qui correspond parfaitement à l'esprit de son entreprise. Etant donné les spécificités de chaque entreprise, il s'agira d'ajouter une partie de paramétrage où l'entreprise pourra modifier le fonctionnement de l'application dépendamment des profils d'utilisateurs d'où la notion d'ERP orienté CRM.

Notre document sera structuré en quatre chapitres :

- La présentation générale ;
- La spécification et l'analyse des besoins ;
- La conception de la solution ;
- Enfin, la présentation de l'application mise en place.

CHAPITRE I : PRESENTATION GÉNÉRALE

Introduction : Ce chapitre est consacré à la présentation de notre entreprise d'accueil, de notre sujet par le biais de son contexte, sa problématique, ses objectifs et enfin de la démarche méthodologique que nous allons emprunter.

I. Présentation de la structure d'accueil [1]

SOPARTECH Développement est une filiale du groupe EXTEL Internationale. Il s'occupe principalement de tout ce qui est développement dans le dit groupe.

a. Société

Fort de 20 ans d'expérience, le Groupe EXTEL est bien ancré dans son rôle d'éditeur de progiciels d'assurance sur le marché européen et est reconnu en tant que tel. D'envergure internationale, le Groupe EXTEL regroupe dans son portefeuille clients des grands comptes de l'assurance européenne et mondiale. Sa gamme de produits et ses services répondent parfaitement aux enjeux du secteur de l'assurance vie et de la finance. EXTEL est aujourd'hui le spécialiste européen de l'édition de progiciels d'assurance. Elle a été Fondée en 1986 par Monsieur Louis-Gérard Brosse et est composée de 4 sociétés :

- **EXTEL sise à Paris ;**
- **EXTEL International sise au Luxembourg ;**
- **IRCI-LOG sise à Paris ;**
- **SOPARTECH Développement sise à Dakar.**

Au travers de leurs différentes activités d'édition de progiciel, de consulting, de prestations de services et de formation, ces 4 entités rassemblent une large gamme de compétences au service du secteur assurantiel. EXTEL a toujours fait le choix d'investir régulièrement pour offrir à ses clients et à ses prospects des solutions nouvelles, toujours avec une approche prudentielle.

b. Stratégie

Le choix en 1999 de la technologie XML pour son premier module FrontOffice montre sa capacité à discerner les technologies porteuses. Trois points sont primordiaux et dictent la ligne de conduite en matière de Recherche et Développement :

- **La pérennité ;**

- **La fiabilité et l'évolution avec les nouvelles technologies ;**
- **La couverture fonctionnelle.**

La stratégie d'EXTEL a été depuis l'origine l'innovation et la sécurité dans les logiciels. C'est sa manière de participer à l'amélioration de la rentabilité de ses clients. Ses techniques de développement assisté par ordinateur et son organisation ont permis une efficacité et une qualité accrues.

Depuis 20 ans, EXTEL déploie tout son talent dans la conception et le développement de solutions innovantes pour l'assurance et la finance.

c. Solutions

Editeur européen majeur de solutions logiciel, EXTEL dispose d'un large panel de solutions métiers afin de toujours mieux répondre aux attentes des acteurs du marché. Quel que soit le domaine métier concerné et les préférences technologiques, EXTEL saura orienter ses clients et ses prospects vers la solution la plus adéquate, avec toujours en arrière fond, une parfaite connaissance de leur métier. Les solutions proposées et dans lesquelles intervient SOPARTECH Développement sont :

- **Sunshine Software ;**
- **Sunshine Life ;**
- **Sunshine Framework ;**
- **MIKADO Framework**

d. Présentation de MIKADO Framework :

Le Framework MIKADO est un outil puissant de création d'application web avec une architecture modulaire. Il propose à l'utilisateur une interface à travers laquelle il peut créer une base de données, un menu et les traitements derrière, gérer le contrôle sur les données, etc. Il donne également la possibilité de faire du reengineering à partir d'applications existantes.

Le Framework comprend :

- Un module qui gère l'espace de travail (Workspace), c'est un composant que nous allons créer au tout début et où nous allons mettre tout ce qui concerne à notre application. A la création du composant, la base de données associée est créée en même temps.
- Un ensemble de développements RAD (Rapid Application Development). C'est une méthode qui se base sur un modèle en spirale et permet d'avoir des itérations rapides.

- Le multithreading est aussi pris en compte de façon efficace. Dans un contexte multiprocesseur, MIKADO permet des exécutions performantes.
- Un module qui permet de tester nos composants même en les construisant.
- Un Gestionnaire de packages pour permettre la distribution du composant dans le contexte d'installation initial et d'upgrades (gestion de versions et releases).
- Un module qui permet de déployer le package sur un serveur cible.
- Un gestionnaire de paramétrage applicatif pour prendre en charge les niveaux de versions des paramètres.
- Un module de génération d'écran JSP sans programmation nommé Screen Designer.

MIKADO s'est inspiré du modèle MVC (Modèle, Vue, Contrôleur) et a été développé en utilisant les outils et technologies J2EE, TOMCAT, XML, ORACLE, HTML, CSS, JavaScript, JQuery, etc.

II. Présentation du sujet

Ce sujet soumis à notre analyse entre dans le cadre de l'amélioration de la communication entre les entreprises et leurs clients. Ainsi, nous présentons tout d'abord le contexte du sujet, puis la problématique, ensuite les objectifs visés et enfin l'état de l'art du CRM et de l'ERP.

a. Contexte

Les entreprises se sont vite rendues compte que pour offrir de meilleurs services à leurs clients, il faut bien les connaître. Il faut avoir une vue d'ensemble sur leurs comportements, interagir avec eux au quotidien, surveiller chaque interaction avec eux et ses impacts, trouver des stratégies pour acquérir de nouveaux clients, etc. Tous ces besoins ont donné naissance à la notion de CRM et ce dernier est aujourd'hui utilisé par beaucoup d'entreprises.

b. Problématique

SOPARTECH développement a essayé de mettre en place une application CRM avec le Framework MIKADO pour des entreprises. Cependant, l'application n'était pas conforme aux besoins des clients. En effet, il y'a beaucoup de problèmes qui se posent :

- L'application n'est pas adaptée à tous les modèles d'entreprises : les utilisateurs de l'application ont besoin de la personnaliser conformément au fonctionnement de leur entreprise ;

- L'ergonomie n'est pas adaptée à la philosophie des CRM ;
- L'utilisateur a du mal à se déplacer facilement entre les différentes informations d'un client ;
- Les pages sont statiques ;
- Il y'a beaucoup de bugs dans l'application.

Pour pallier ces problèmes, nous nous sommes fixés quelques objectifs.

c. Objectifs

i. Objectif global

L'objectif de notre travail est de créer une application de gestion de la relation client en nous inspirant de l'existant. L'application doit être conforme à la philosophie applicative des CRM, et l'entreprise doit pouvoir la personnaliser en adéquation avec son fonctionnement.

ii. Objectifs spécifiques

Les futurs utilisateurs de notre application doivent pouvoir :

- Disposer d'une source unique d'informations
- Gérer et avoir une vue d'ensemble sur les informations concernant un client
- Rechercher rapidement une information
- Paramétrer les vues, la disposition des éléments dans l'application, les formulaires, les accès en fonction des profils de l'entreprise
- Gérer les agendas, les emails
- Disposer d'une représentation synthétique des informations les plus pertinentes et indicateurs de performance.

d. Etat de l'art du CRM et de l'ERP

i. Définition

Le CRM ou gestion de la relation client (Customer Relationship Management) est une stratégie de gestion des relations et interactions d'une entreprise avec ses clients ou clients potentiels. Un système CRM aide les entreprises à interagir en permanence avec les clients, à rationaliser leurs processus et à améliorer leur rentabilité.

L'ERP (Enterprise Resource Planning) ou progiciel de gestion intégré en français est un progiciel qui permet de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble de ses fonctions, dont la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais

aussi la vente, la distribution, l'approvisionnement et le commerce électronique [2]. Dans notre cas, il s'agit d'ERP orienté dans la gestion des clients.

ii. Fonctionnalités de base d'un CRM

Parmi les principales fonctionnalités d'un système de CRM, nous comptons généralement :

- La gestion des contacts : Les derniers renseignements relatifs aux clients (depuis leurs coordonnées jusqu'aux conversations de services) sont facilement accessibles et modifiables.
- La gestion des potentiels clients : Ce système permet aux utilisateurs de suivre les activités, les tâches et les objectifs liés au pipeline¹, depuis le prospect² jusqu'à la conversion.
- Les prévisions des ventes : Les rapports de prévisions permettent aux commerciaux d'avoir une meilleure visibilité sur leurs pipelines, de qualifier des pistes commerciales avec une plus grande précision et de savoir s'ils atteignent leurs objectifs. Les responsables des ventes peuvent utiliser les rapports pour encourager et gérer leurs collaborateurs.
- Suivi et intégration de messagerie électronique (Outlook et Gmail) : La synchronisation instantanée des clients de messagerie avec le système de CRM permet aux employés d'avoir une vue d'ensemble de leurs clients et potentiels clients sans devoir se connecter à plusieurs systèmes. Les calendriers et les contacts sont disponibles sur n'importe quel appareil et les e-mails peuvent être créés et gérés depuis un workflow unique.
- Analyses basées sur des tableaux de bord : Les informations sont regroupées et présentées dans un tableau de bord intuitif, clair et personnalisable selon les priorités de chacun [1].

Comme exemples de solutions CRM, nous avons noCRM, Axonaut, Hubspot, Salesforce, etc.

III. Méthode de travail

Une méthodologie de gestion de projet est un ensemble d'étapes à suivre pour bien réaliser un projet. Parmi les phases clés de ce processus, nous avons l'analyse des besoins et la conception. Nous allons par la suite bien élucider ces termes, présenter quelques méthodes de gestion de projet et en choisir une.

a. Définition de concepts

i. L'analyse

¹ Un pipeline est une représentation visuelle des prospects et du stade où ils se trouvent dans le processus d'achat.

² Un prospect est un potentiel client, une piste commerciale

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc. C'est dans cette phase qu'on répond à la question « que fait le système ? ». Elle peut être découpée en 3 phases :

- La définition des besoins : Il s'agit d'identifier les acteurs et les cas d'utilisation, de structurer le modèle, et d'identifier les autres exigences.
- La capture des besoins : elle consiste à collecter des informations (interviews, lecture de documentation) nécessaires à la compréhension du domaine et du problème posé. A ce niveau il s'agit de restituer les besoins dans un langage compréhensible par le client et de procéder à l'identification, à la structuration et à la définition d'un dictionnaire.
- La spécification des besoins : dans cette phase il sera question d'aller à un niveau de spécification plus détaillé voire même plus formel des besoins. Elle sera d'une grande utilité pour le client mais aussi pour le développeur. A la fin de cette phase d'analyse, on aura un modèle conceptuel, outil fondamental lors de la phase de conception.

ii. La conception

Phase menée à la suite de l'analyse des besoins, la conception a comme objectifs de répondre à la question « comment faire le système ? » et de décomposer de façon modulaire le système à mettre en place. La conception définit l'architecture du logiciel et par la même occasion chaque constituant du logiciel (Informations traitées, traitements effectués, résultats fournis, contraintes à respecter). A la fin de cette phase, un modèle logique utilisable à la phase d'implémentation est produit.

b. Classification des méthodologies de gestion de projet [3]

Il existe plusieurs méthodes de travail pour mettre en place des logiciels. Nous allons vous présenter la méthode traditionnelle et la méthode agile.

1. Les méthodologies traditionnelles

Les méthodes classiques sont aussi appelées « en cascade » car chaque étape doit être terminée pour passer à la suivante.

En appliquant cette méthodologie, l'équipe projet suit le cahier des charges à la lettre et travaille sur la totalité du projet jusqu'à sa livraison. Il n'y a pas d'interaction avec le client qui recevra son projet une fois que celui-ci est terminé.

Au final, le client est souvent déçu car ses attentes ont évolué ou le contexte a changé. Des imprévus sont survenus, et des ajustements doivent être effectués, ce qui implique des retards et des coûts supplémentaires.

2. Les méthodologies agiles

Plus efficaces et moins rigides que les méthodes classiques, les méthodes agiles placent les besoins du client au centre des priorités du projet. Elles offrent une plus grande flexibilité et une meilleure visibilité dans la gestion du projet, ce qui permet à l'équipe d'être plus réactive aux attentes du client.

Le projet est ainsi découpé en mini-projets, chacun nécessitant la validation du client pour passer au suivant. Le dialogue avec le client est privilégié, les retours et les ajustements sont possibles. On prend davantage en considération l'évolution des besoins du client.

c. Choix d'une méthodologie de gestion de projet

Le choix de notre méthodologie de gestion de projet devra être motivé aussi bien par le sujet que par le cadre dans lequel nous travaillons.

Dans ce projet, nous évoluons dans un environnement où les besoins du client n'ont pas été définis de manière claire dès le début du projet. En plus, ces besoins changent constamment du fait de la présence du client dans le processus. De ce fait nous serions tentés de nous pencher vers une méthode agile qui correspond à ce cadre de développement.

Pour mener à bien notre projet, il serait pertinent de procéder par de petites itérations, les améliorer, les réadapter jusqu'à ce que ça soit conforme aux besoins du client. Cela nous fait penser au Framework Scrum que nous allons utiliser pour notre projet.

Pour apporter une compréhension rapide de notre système avec une bonne modélisation, nous allons utiliser le langage UML.

Par ailleurs, la valeur n'est pas dans la méthode mais dans les personnes qui la mettent en œuvre.

i. Présentation de Scrum [4]

Scrum est un cadre de développement de produits logiciels complexes se basant sur la méthodologie agile. Le terme Scrum (qui signifie mêlée) apparaît pour la première fois en 1986 dans une publication de Hirotaka Takeuchi et Ikujiro Nonaka qui décrit une nouvelle approche plus rapide et flexible pour le développement de nouveaux produits.

1. Principe

L'approche Scrum suit les principes de la méthodologie Agile, c'est-à-dire l'implication et la participation active du client tout au long du projet. Scrum se compose de plusieurs éléments fondamentaux :

- Des rôles,
- Des événements,
- Des artefacts,
- Des règles.

Il s'agit d'une approche empirique (c'est-à-dire qui se base sur l'expérience), dynamique et participative de la conduite du projet. Au rugby, la mêlée est une phase indispensable car elle permet au jeu de repartir sur d'autres bases. Même chose pour Scrum : l'équipe se réunit quotidiennement lors d'une réunion de synchronisation, appelée mêlée quotidienne, afin de suivre l'avancement du projet.

2. Répartition des rôles

L'équipe Scrum est autoorganisée et pluridisciplinaire, c'est-à-dire qu'elle choisit la meilleure façon d'accomplir son travail et qu'elle possède toutes les compétences nécessaires à l'accomplissement du projet. La flexibilité, la créativité et la productivité de l'équipe sont ainsi optimisées.

L'équipe Scrum se compose de :

- Un Scrum Master : responsable de la compréhension, de l'adhésion et de la mise en œuvre de la méthode Scrum qu'il maîtrise parfaitement
- Un Product Owner ou propriétaire du produit
- Une équipe de développement

3. Les différents événements

La vie d'un projet Scrum est rythmée par un ensemble de réunions définies avec précision et limitées dans le temps : le Sprint, La planification d'un Sprint, la revue de Sprint, la rétrospective du sprint et la mêlée quotidienne.

ii. Application de Scrum à notre projet

Après avoir présenté Scrum, nous détaillons la répartition de l'équipe projet conformément à Scrum :

Le Product Owner : c'est le directeur de l'entreprise SOPartech Développements ;

Le Scrum Master : un des employés de l'entreprise ;

L'équipe de développement : elle est constituée de 2 développeurs full stack.

La durée des sprints est de 3 semaines. Des revues et des rétrospectives sont organisées à la fin de chaque sprint ainsi que des réunions hebdomadaires pour voir l'avancement du projet.

Pour la modélisation du système, nous allons utiliser le langage UML.

iii. Présentation du langage de modélisation UML [5]

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

Les versions d'UML 1.x proposaient neuf (09) diagrammes. UML 2.0 en a rajouté quatre. Ces treize types de diagrammes représentent autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

❖ **Diagrammes structurels ou diagrammes statiques (UML Structure) :**

Il s'agit des diagrammes qui permettent de représenter le système physiquement. Ils sont au nombre de six :

- Diagramme de classes (Class diagram) ;
- Diagramme d'objets (Object diagram) ;
- Diagramme de composants (Component diagram) ;
- Diagramme de déploiement (Deployment diagram) ;
- Diagramme de paquetages (Package diagram) rajouté par UML 2.0 ;
- Diagramme de structures composites (Composite structure diagram) rajouté par UML 2.0.

❖ **Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior) :** Il s'agit des diagrammes montrant le fonctionnement du système à concevoir. Ils sont au nombre de 7 :

- Diagramme de cas d'utilisation (Use case diagram) ;
- Diagramme d'activités (Activity diagram) ;
- Diagramme d'états-transitions (State machine diagram) ;
- Diagrammes d'interaction (Interaction diagram) :
 - Diagramme de séquence (Sequence diagram) ;
 - Diagramme de communication (Communication diagram) ;
 - Diagramme global d'interaction (Interaction overview diagram) rajouté par UML 2.0 ;
 - Diagramme de temps (Timing diagram) rajouté par UML 2.0.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états transitions. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

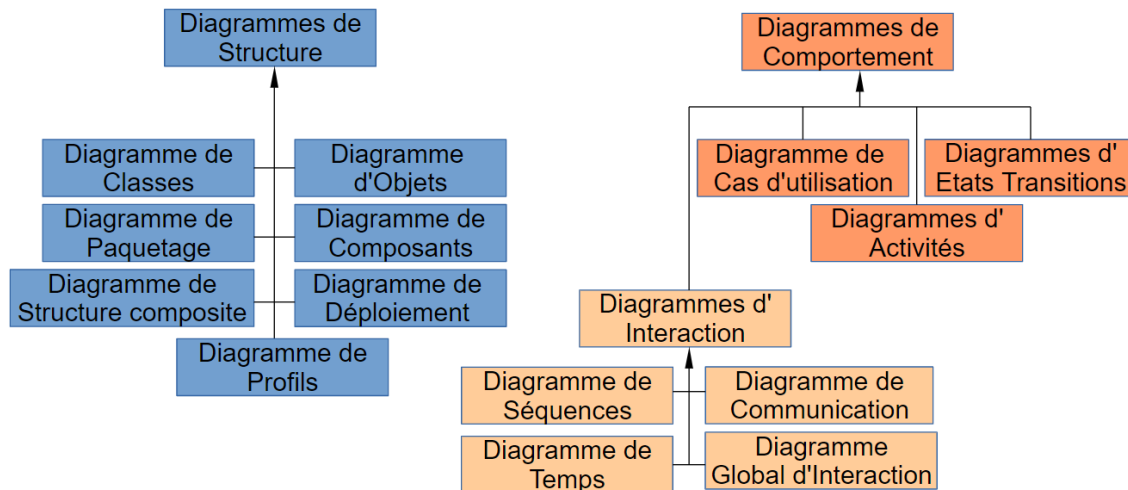


Figure 1: Les treize diagrammes d'UML

Le choix d'une démarche et d'un langage de modélisation étant fait, nous pouvons maintenant passer à la spécification et l'analyse des besoins proprement dits de l'application.

Conclusion : Dans ce chapitre nous avons présenté la structure d'accueil ainsi que le sujet du mémoire. Nous avons aussi présenté la méthode de gestion de projet choisie. Dans le chapitre suivant nous allons passer à la spécification et à l'analyse des besoins.

CHAPITRE 2 : SPECIFICATION ET ANALYSE DES BESOINS

Introduction : Ce chapitre est consacré à l'étude de l'existant, l'analyse et la modélisation de notre système. Ces étapes vont nous permettre de mieux comprendre le fonctionnement du système et de savoir comment le faire évoluer.

I. Etude de l'existant

SOPartech Développement a mis en place une application CRM avec le Framework Mikado.

a. Présentation de l'application

L'application Alias CRM a été conçue pour permettre aux entreprises de gérer et centraliser les données de leurs clients. Parmi ses fonctionnalités, nous retrouvons :

- La création de clients physiques avec la possibilité de consulter toutes ses informations à savoir l'état civil, l'adresse, les informations fiscales, le revenu, etc.
- La création de clients moraux comme les établissements avec la possibilité de consulter les détails de l'établissement, les bénéficiaires, le registre de commerce, l'adresse, etc.
- L'envoi de mails, le listing et la consultation.
- La gestion (ajout, modification, consultation, recherche, suppression) des fiches revenus, registres commerciaux, informations fiscales, pièces administratives, invalidités, assurances santé, etc.

Nous vous présentons quelques captures de l'application :

La figure 2 représente la page d'accueil de l'application Alias CRM. Elle montre l'ensemble des fonctionnalités disponibles dans l'application.

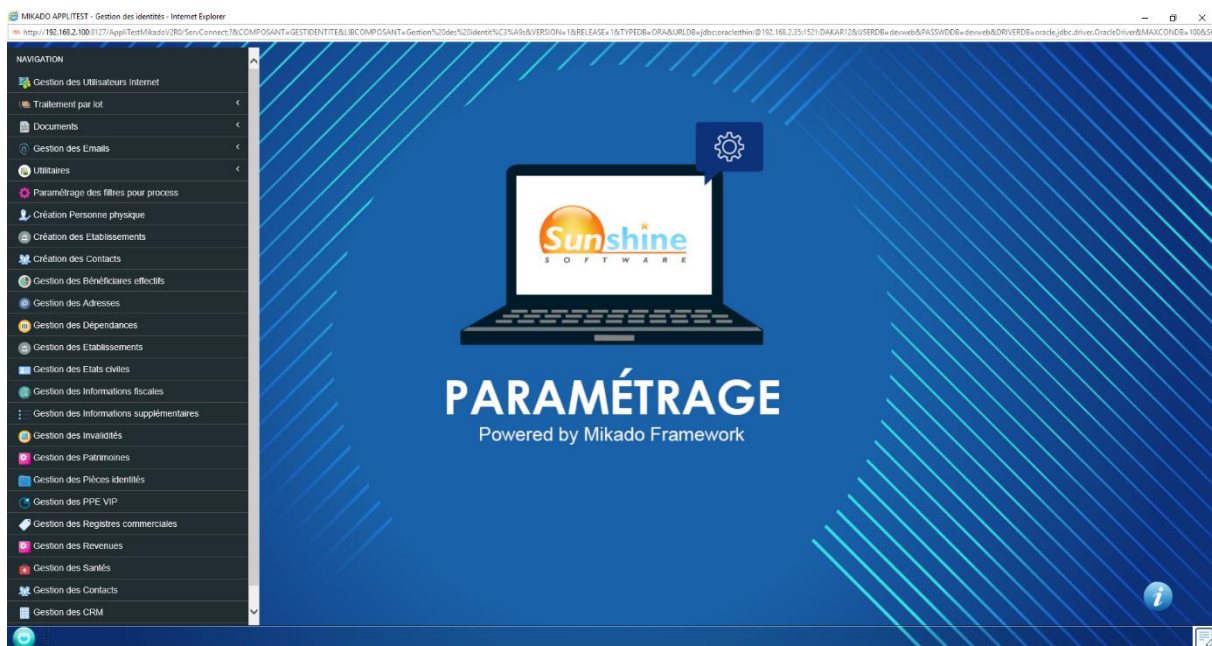


Figure 2 : Page d'accueil de l'application Alias CRM

La figure 3 représente la liste des clients enregistrés dans la base de données. Il y'a également un filtre disponible pour rechercher des clients bien déterminés.

Création Personne physique

Liste physique Etat civile

Critère de recherche

N° CRM Date Effet des informations Entre le Et le

Civilité Nom

Référence INSEE Prénom

De 1 à 7

	N° CRM	Civilité	Nom	Prénom	Référence INSEE	Date Effet des informations
<input type="checkbox"/>	1	<input type="text"/>	Sabrina	Saber	275127500942403	13/08/2018
<input type="checkbox"/>	3	<input type="text"/>	Dupon	Dup	172029932822908	14/08/2018
<input type="checkbox"/>	5	<input type="text"/>	George	George	168127500635273	14/08/2018
<input type="checkbox"/>	6	<input type="text"/>	Isabelle	Belaire	272061600123539	14/08/2018
<input type="checkbox"/>	7	<input type="text"/>	Julien	Jule	172067500154331	14/08/2018
<input type="checkbox"/>	8	<input type="text"/>	Fabien	Bee	182087501235711	14/08/2018
<input type="checkbox"/>	9	<input type="text"/>	Juliette	June	268107500354384	14/08/2018

Figure 3 : Liste des clients

La *figure 4* représente la page d'ajout d'une identité physique. Elle renseigne en même temps la fiche Etat civil. Nous avons ensuite la possibilité de cliquer sur les icônes en haut de page pour renseigner les autres fiches comme les fiches Revenu, Informations fiscales, etc.

Création Personne physique

PHSAISIETATCIV

[Retour à la liste](#)

Etat civile

N° CRM Type d'identité 1 - Personne physique Date Effet des informations 30/06/2021

Sexe

Civilité Nom Prénom

Nom crypté Nom naissance Prénom2

Né(e) le à Lieu de naissance du Commune de naissance

Nationalité

Situation Familiale Indicateur date reconstitué

Référence INSEE

Informations du Décès

Date de Décès Date de Connaissance du Décès

Information cause Décès

Présumé Décès

ANNULLER VALIDER

Figure 4 : Ajout d'un client de l'application Alias CRM

Si on veut modifier ou consulter une fiche déjà rempli, on a une interface comme celle de la figure 5. On choisit d’abord l’identité sur la liste, on pourra ensuite modifier ou consulter sa fiche.

The screenshot shows a web application titled "Gestion des Registres commerciales". It features a search section with the following fields:

- N° CRM: []
- N° Siren: []
- Dénomination sociale: []
- Date Effet des informations: Entre le [] Et le []
- RECHERCHER button

Below the search section is a table with the following data:

	N° CRM	N° Siren	Dénomination sociale	Date Effet des informations
<input type="checkbox"/>	2	254399911	SAS Bar	13/08/2018
<input type="checkbox"/>	10	468735136	Bar	14/08/2018
<input type="checkbox"/>	11	327837435	Bar	14/08/2018

Figure 5 : Liste des registres de commerce des identités morales de Alias CRM

La figure 6 montre l’interface d’envoi d’email avec la possibilité de joindre des pièces. Il est également possible d’envoyer des emails groupés. Cette fonctionnalité est prise en compte par les traitements par lots. Ces derniers sont paramétrés au moment de la création de l’application à travers le Framework MIKADO et permettent de faire des traitements en même temps pour plusieurs clients.

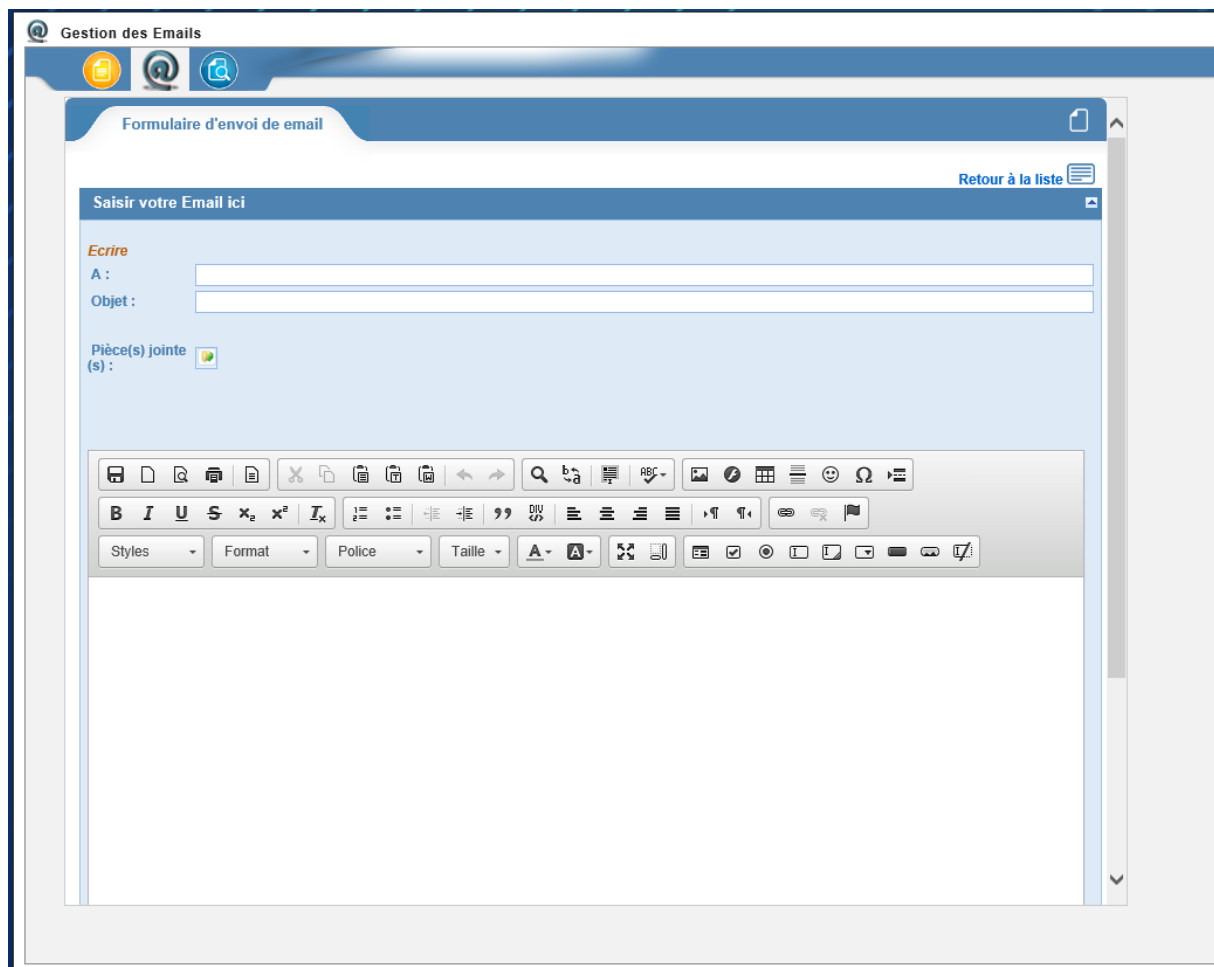


Figure 6 : Envoi d'emails dans l'application Alias CRM

b. Etude critique de l'existant

Dans l'application Alias CRM, nous avons remarqué quelques bugs, des traitements qui ne fonctionnent pas correctement au niveau de:

- L'ajout d'une personne politiquement exposée (PPE VIP)
- L'ajout de pièces administratives
- L'ajout de patrimoine
- L'ajout d'une invalidité
- L'ajout d'une personne physique
- L'ajout d'une dépendance
- L'ajout d'une adresse
- L'ajout d'un contact
- La création et l'envoi de mails
- Etc.

Ces erreurs sont pour la plupart liées au paramétrage des tables physiques et lien entre ces tables.

Du point de vue organisation des écrans, force est de remarquer qu'il est difficile de se déplacer entre les différentes informations des personnes et des établissements. Les écrans ne sont pas modernes et assez adaptés à la philosophie des CRM. Également, Il est pertinent dans un CRM de transformer les données des clients en informations utiles et les présenter sous forme de graphes pour mieux aider à la prise de décision.

En plus, il est impossible de personnaliser l'application, on est obligé de créer une application pour chaque entreprise désirant utiliser le CRM.

Cette dernière, qui n'est pas responsive, propose un seul modèle de présentation des données qui n'est pas forcément adapté à toutes les entreprises : les écrans générés avec Screen Designer sont statiques. Si on veut ajouter un nouveau champ dans une table il va falloir redessiner les écrans et redéployer l'application. Ce sont les limites du Framework MIKADO. Nous allons ainsi l'associer avec d'autres technologies pour avoir une application CRM conforme aux exigences des clients.

Capturons donc à nouveau les besoins en commençant par redéfinir les acteurs du système.

II. Acteurs et Rôles

L'application que nous allons concevoir est destinée aux entreprises notamment les entreprises d'assurance utilisant déjà les solutions de SOPartech développements comme le Progiciel SUNSHINE, le Framework MIKADO, etc.

Les acteurs du système sont principalement de 2 types :

- **Les administrateurs** : ils sont des membres de l'entreprise qui se chargent de gérer, entre autres, le paramétrage de l'application, des utilisateurs, leurs profils, etc.
- **Le responsable CRM ou un autre profil de l'entreprise** : il revient à l'entreprise de déterminer qui fait quoi dans l'application. Chaque profil a un rôle et les privilèges qui lui sont associés. Le responsable CRM sera chargé, entre autres, d'enregistrer les clients et de faire le suivi.

III. Les besoins fonctionnels

Les fonctionnalités de notre système sont définies en fonction des profils d'utilisateurs donc une authentification sera nécessaire.

Après s'être authentifié, l'administrateur pourra accéder à l'espace de paramétrage où il pourra :

- Gérer les utilisateurs et leurs profils

- Gérer des tables physiques avec la possibilité d'ajouter, modifier, consulter ou supprimer des tables ou champs.
- Paramétrer le menu
- Créer, modifier, consulter ou supprimer des listes de valeurs pour les champs dont les valeurs font partie d'une énumération.
- Définir des liens et dépendances entre les tables de la base de données
- Paramétrer le tableau de bord
- Organiser l'affichage des fiches clients en fonction du profil.

L'utilisateur simple, qui est un profil de l'entreprise, pourra :

- Gérer ses clients
- Ajouter, modifier, consulter et supprimer les fiches des clients
- Rechercher rapidement une information
- Gérer l'envoi et la réception d'emails avec la possibilité de choisir des modèles d'emails prédéfinis
- Gérer son agenda : ajout, suppression ou modification d'événements avec la possibilité d'avoir des rappels
- Consulter le tableau de bord.

IV. Les besoins non fonctionnels

Les besoins non fonctionnels représentent le comportement et la performance que notre solution doit avoir.

En effet, en plus d'être fiable et sécurisée, l'application CRM que nous devons mettre en place doit avoir une très bonne ergonomie adaptée à la philosophie des CRM. L'utilisateur doit pouvoir se déplacer facilement entre les différentes fiches des clients. Bien qu'il y ait beaucoup d'informations à stocker, l'application doit être fluide. Elle doit également être facilement adaptable à n'importe quel type d'entreprise.

V. Modélisation

La réalisation de notre application CRM, comme toute autre application, nécessite une spécification rigoureuse des fonctionnalités. C'est à cet effet que nous vous présentons deux types de diagramme à

savoir le diagramme de cas d'utilisation et le diagramme de séquence. Il y'aura également quelques fiches textuelles pour une vue plus détaillée des cas d'utilisation.

a. Présentation des fonctionnalités

Dans cette partie, nous utiliserons le diagramme de cas d'utilisation pour présenter les fonctionnalités les plus importantes du système.

Nous avons d'abord la gestion de l'espace de paramétrage de notre application. Dans cet espace, l'administrateur pourra changer le fonctionnement de l'application en ajoutant de nouvelles tables dans la base de données, en changeant par exemple la couleur de certaines vues de l'application, etc.

Nous avons également une partie qui gère les informations des clients. L'utilisateur pourra avoir une vue d'ensemble sur les données clients, les manipuler, etc.

La *figure 7* montre plus en détail le découpage en package de notre application.

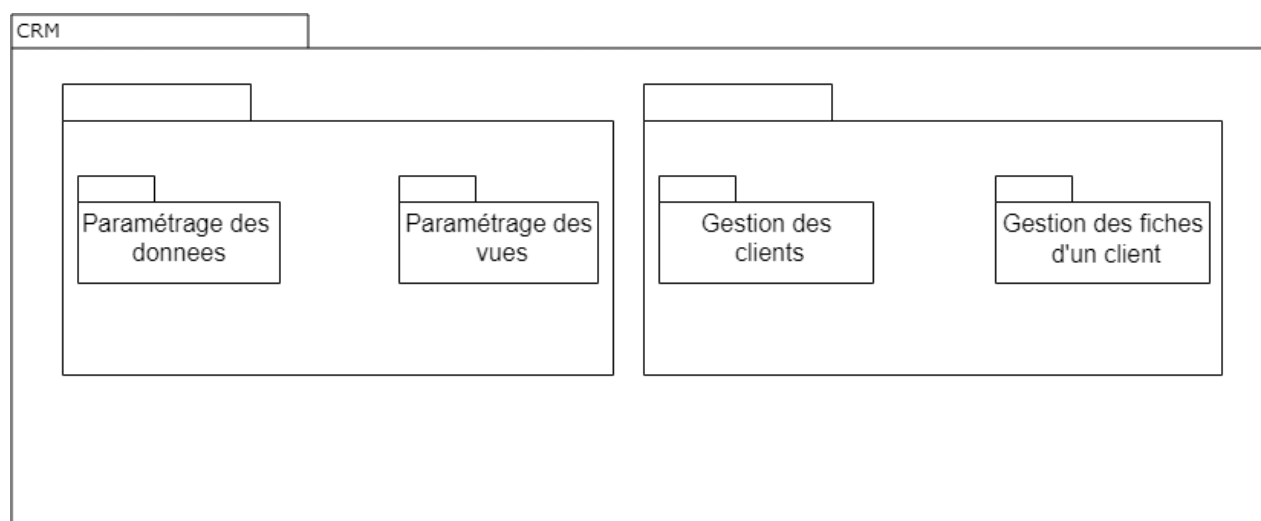


Figure 7 : Diagramme de package de notre application

- **La gestion du paramétrage**

C'est l'espace qui va permettre à l'entreprise de personnaliser le CRM. La *figure 8* montre les paramètres qu'on peut faire sur l'application. La *figure 9* quant à elle, montre les paramètres qu'on peut effectuer sur les données de l'application.

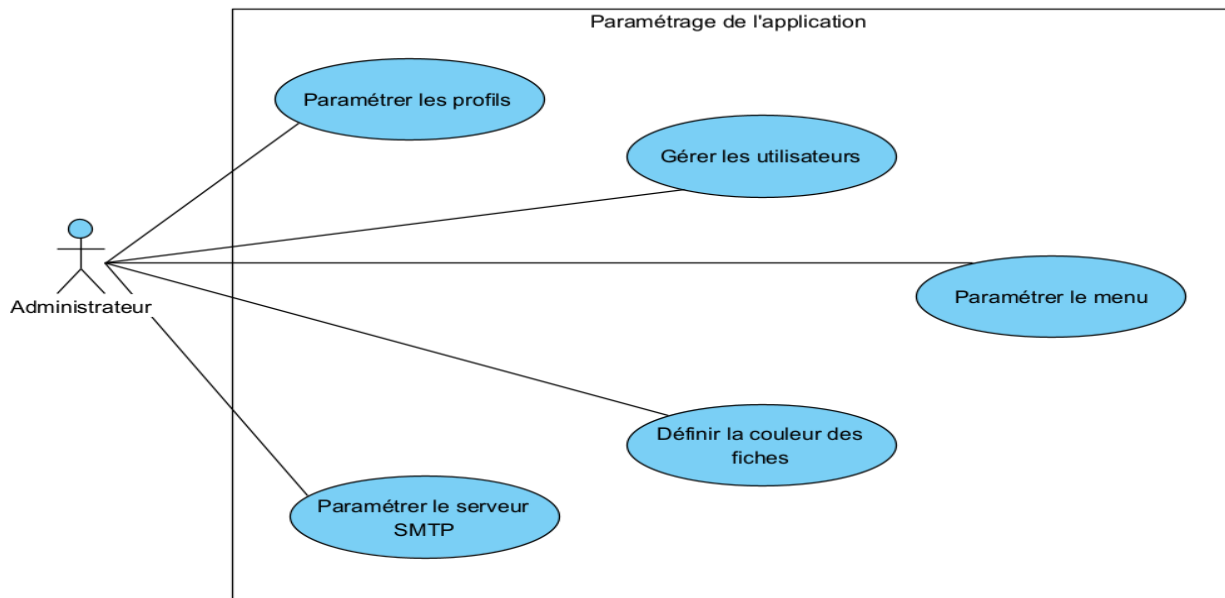


Figure 8 : Diagramme de cas d'utilisation du paramétrage de l'application

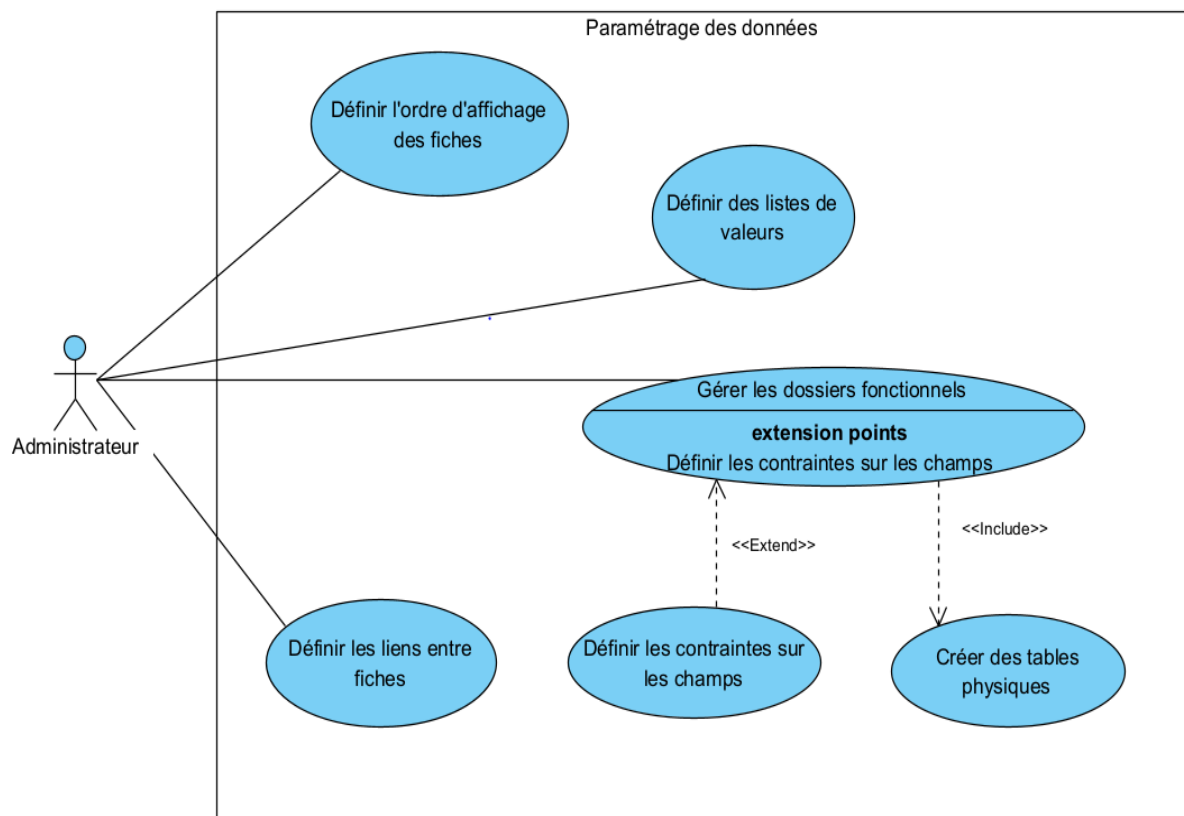


Figure 9 : Diagramme de cas d'utilisation du paramétrage de données

- La gestion des clients :

Dans notre application, l'utilisateur aura la possibilité de gérer les informations de ses clients.

La *figure 10* présente les traitements qu'on peut effectuer sur les clients. La *figure 11* montre les fonctionnalités relatives aux fiches d'information d'un client.

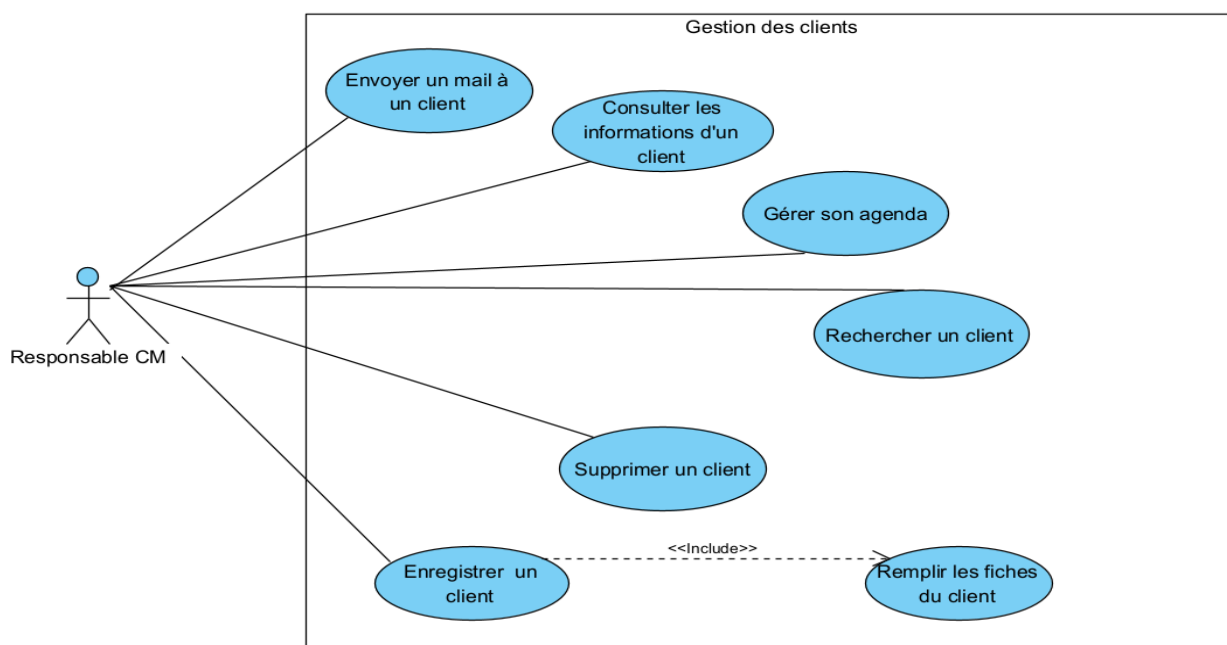


Figure 10 : Diagramme de cas d'utilisation de la gestion des clients

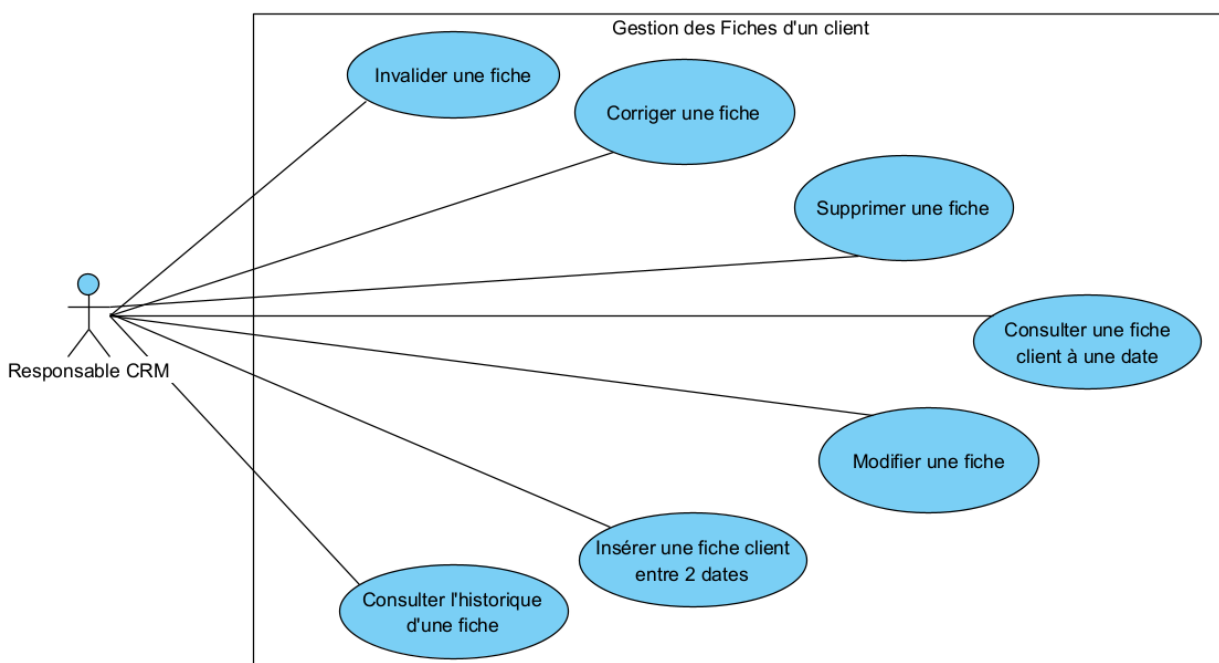


Figure 11 : Diagramme de cas d'utilisation de la gestion des fiches d'un client

Par la suite, nous allons présenter en détails les fonctionnalités « Enregistrer un client », « modifier une fiche » et « Gérer les dossiers fonctionnels » en utilisant une description graphique puis textuelle.

b. Description des fonctionnalités

Pour la description des fonctionnalités, nous utiliserons des diagrammes de séquences accompagnés de fiches textuelles.

Les figures 12 et 13 représentent respectivement la description graphique et textuelle de l'enregistrement d'un nouveau client

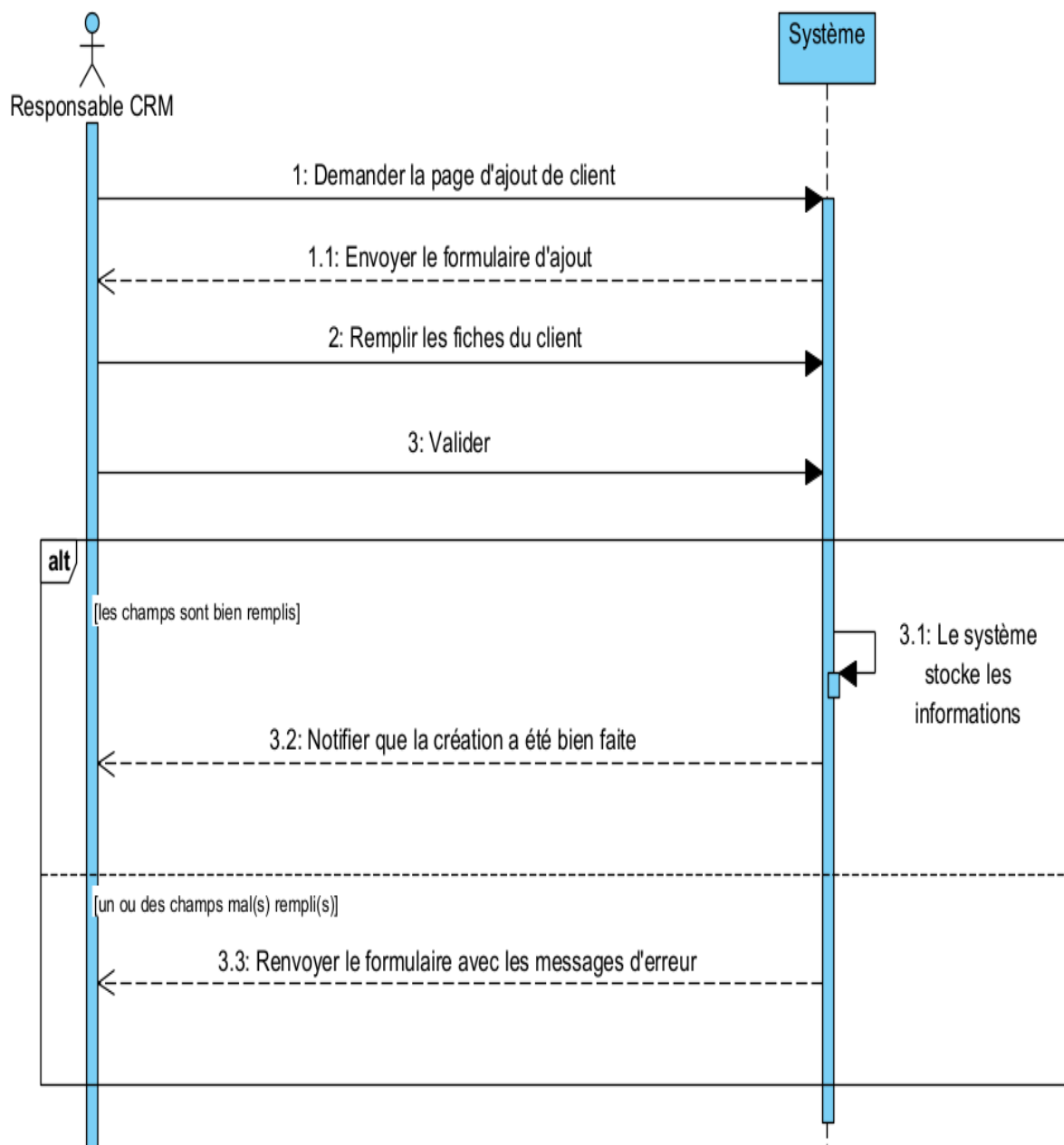


Figure 12 : Diagramme de séquence de l'enregistrement d'un client

Titre	Enregistrer un client
Objectif	Le responsable CRM veut ajouter un nouveau client dans sa base de données
Acteur(s)	Le responsable CRM
Précondition	S'authentifier
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable CRM demande la page d'ajout de client 2. Le système lui envoie le formulaire d'ajout 3. Le responsable CRM remplit les fiches du client 4. Le responsable CRM soumet le formulaire 5. Le système stocke les informations 6. Le système notifie que la création a été bien faite
Scénario alternatif	SA1- Après 4 : s'il y'a des champs mal renseignés 4.1- Le système renvoie le formulaire avec les messages d'erreur.
Postcondition	Le client a été bien enregistré.

Figure 13 : Fiche textuelle de l'enregistrement d'un client

Les figures 14 et 15 décrivent le procédé pour modifier une fiche d'information d'un client.

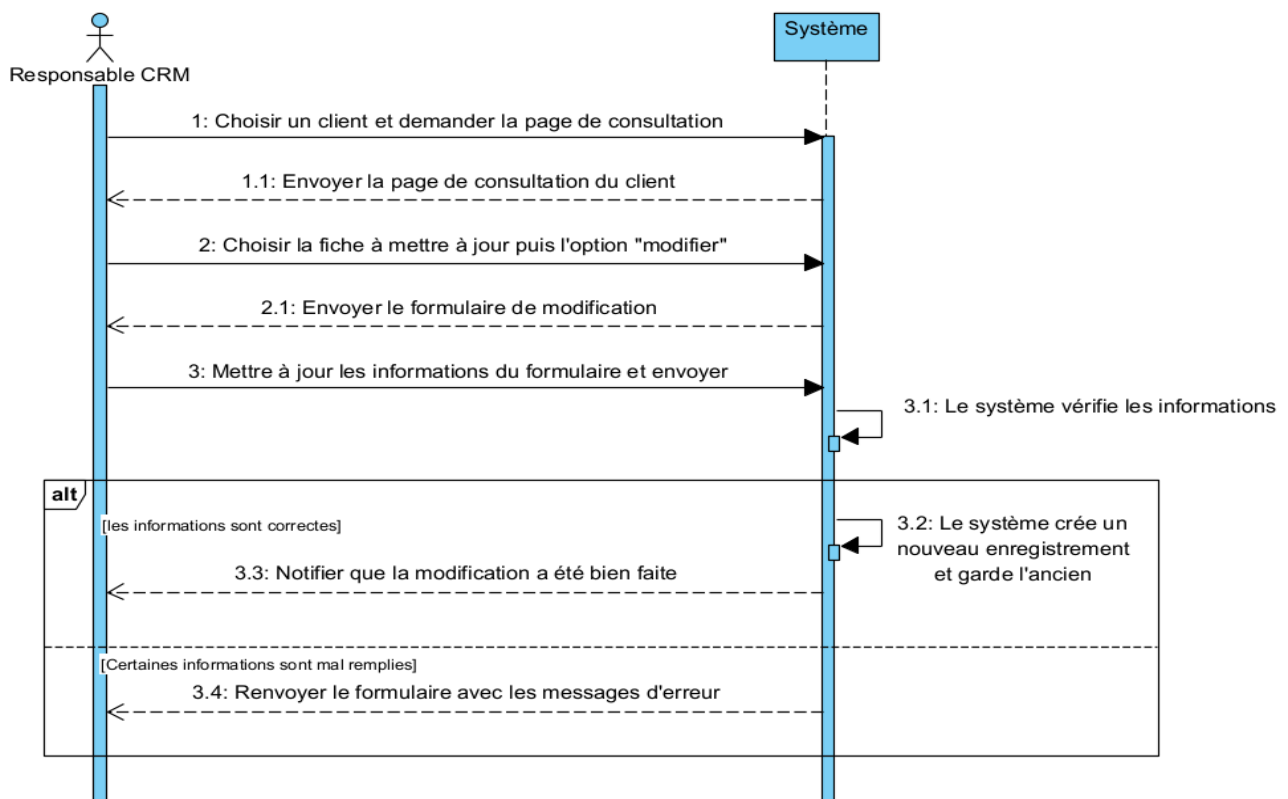


Figure 14 : Diagramme de séquence de la modification d'une fiche client

Titre	Modifier une fiche
Objectif	Le responsable CRM veut mettre à jour les informations d'une fiche
Acteur(s)	Responsable CRM
Précondition	S'authentifier, fiche existante
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable CRM choisit un client et demande la page de consultation 2. Le système lui envoie la page de consultation du client 3. Le responsable CRM choisit la fiche à mettre à jour puis l'option « modifier » 4. Le système lui envoie le formulaire de modification 5. Le responsable CRM met à jour les informations 6. Le responsable CRM soumet le formulaire 7. Le système vérifie les informations 8. Le système crée un nouvel enregistrement et garde l'ancien 9. Le système notifie que la modification a été bien faite
Scénario alternatif	SA1- Après 7 : s'il y'a des champs mal renseignés 7.1- Le système renvoie le formulaire avec les messages d'erreur
Postcondition	La fiche a été bien modifiée.

Figure 15 : Fiche textuelle de la modification d'une fiche client

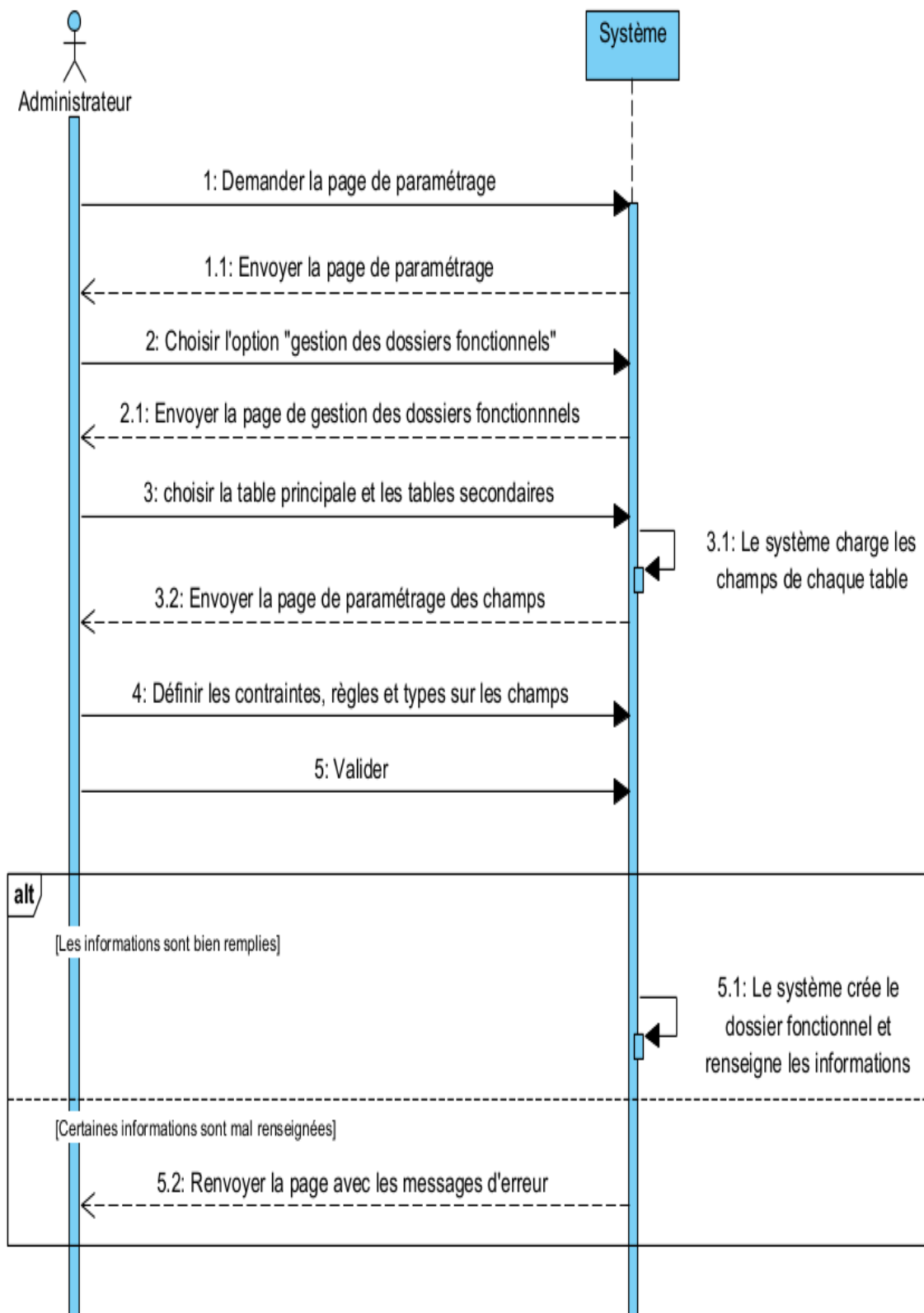


Figure 16 : Diagramme de séquence de la gestion du dossier fonctionnel

Les figures 16 et 17 montrent plus en détail comment gérer les dossiers fonctionnels.

Titre	Gérer les dossiers fonctionnels
Objectif	L'administrateur veut alimenter les dossiers fonctionnels.
Acteur(s)	Administrateur
Précondition	S'authentifier, créer les tables physiques
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande la page de paramétrage. 2. Le système lui envoie la page de paramétrage. 3. L'administrateur choisit l'option « gestion des dossiers fonctionnels ». 4. Le système lui envoie la page concernée. 5. L'administrateur choisit la table principale et les tables secondaires. 6. Le système charge les champs de chaque table. 7. Le système envoie la page de paramétrage des champs. 8. L'administrateur définit les contraintes, règles et types des champs. 9. L'administrateur valide les informations. 10. Le système enregistre les informations.
Scénario alternatif	SA1- Après 9 s'il y'a des champs mal remplis 9.1- Le système lui renvoie la page avec les messages d'erreur.
Postcondition	Le dossier fonctionnel a été bien renseigné.

Figure 17 : fiche textuelle de la gestion du dossier fonctionnel

Pour donner un peu plus de sens à la définition des concepts de notre application, nous présentons des diagrammes de classe d'analyse.

c. Diagramme de classe

Un diagramme de classe représente la structure globale d'une application. Pour mettre en place notre ERP orienté CRM, nous proposons 3 diagrammes de classe.

Nous représentons d'abord tout ce qui concerne les informations personnelles d'un client.

Ensuite, nous nous intéressons aux informations sur ses biens et sa santé.

Enfin, nous décrivons les différentes classes relatives aux paramétrages de l'application c'est-à-dire les classes qui représentent les tables, leurs champs, etc.

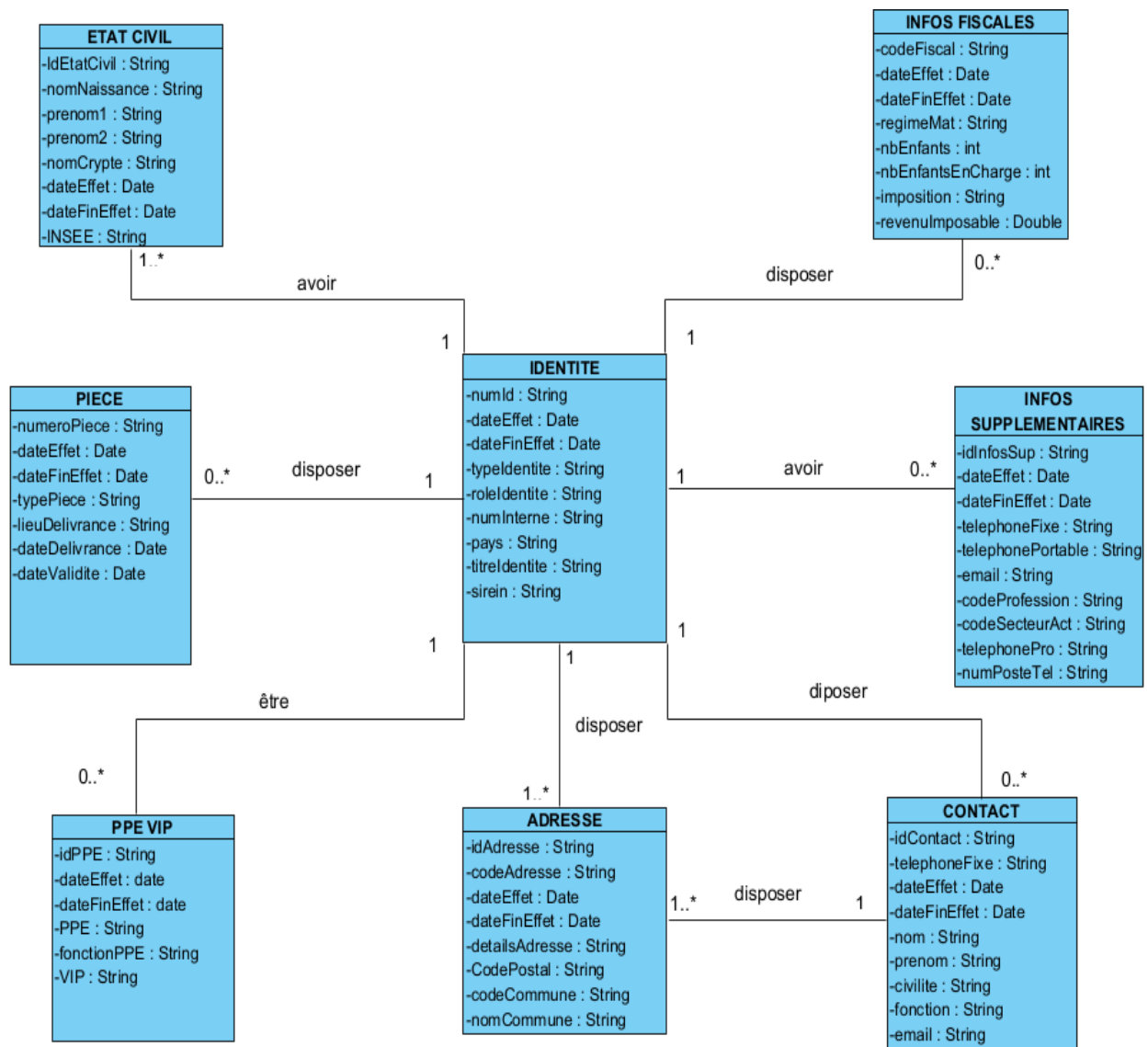


Figure 18: Diagramme de classe des informations personnelles du client

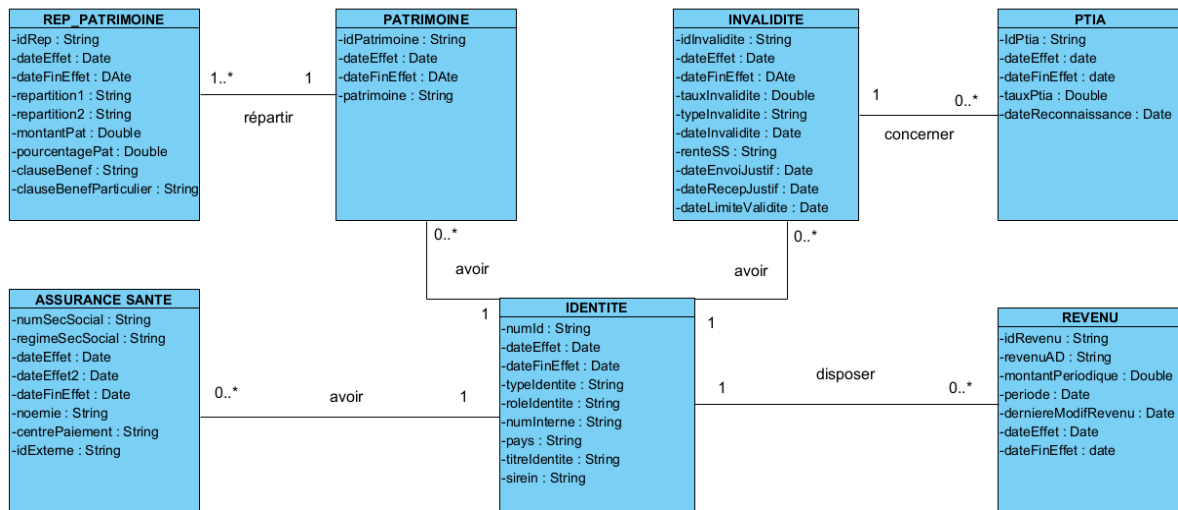


Figure 19 : : Diagramme de classe des informations sur les biens et la sante du client

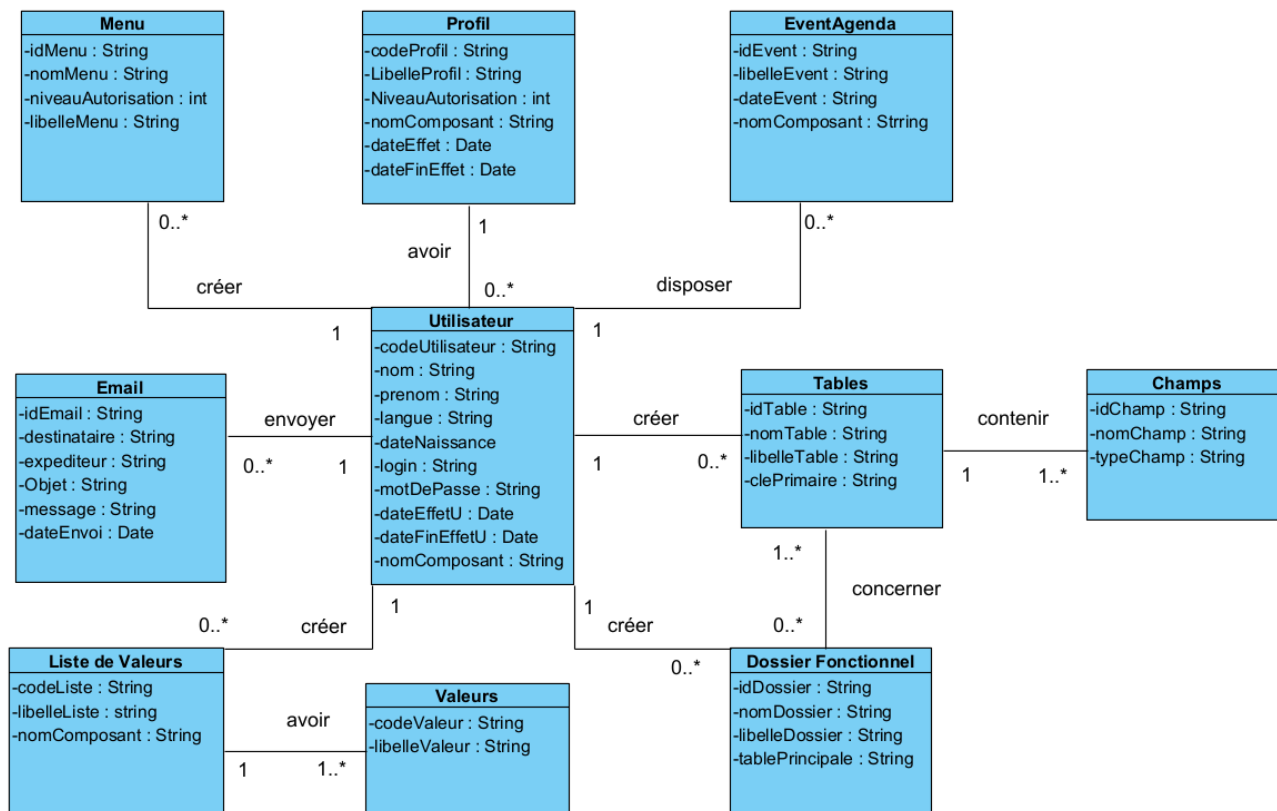


Figure 20 : Diagramme de classe du paramétrage de l'application

VI. Gestion de projet

Maintenant que la spécification et l'analyse des besoins de notre système ont été faites, nous allons, à partir des fonctionnalités identifiées, construire notre Backlog Product conformément à la méthode Scrum. Il sera composé de :

- User stories : une description du contenu d'une fonctionnalité
- Epic : une tâche qui peut être subdivisée en plusieurs user stories
- Sprint : une itération de courte durée décomposant le processus de développement

Nous allons ainsi diviser notre travail en 3 sprints.

Pour le premier Sprint, il s'agira de mettre en place des interfaces graphiques ergonomiques, intuitives et adaptées à la philosophie des CRM.

Pour le deuxième Sprint, nous allons développer notre backend concernant la gestion des clients et l'associer à nos interfaces au fur et à mesure.

Pour le troisième Sprint, nous nous focalisons sur le backend de l'espace de paramétrage des données et des vues.

User Stories	Epic	Priorité	Sprint	Début Sprint	Fin Sprint
Interfaces graphiques de la gestion des clients	Interfaces Graphiques	Haute	1	03/05/2021	24/05/2021
Interfaces graphiques du tableau de bord, de la gestion de l'agenda et des emails	Interfaces Graphiques	Haute	1		
Interfaces graphiques de l'espace de paramétrage des données	Interfaces Graphiques	Haute	1		
Interfaces graphiques de l'espace de paramétrage des vues	Interfaces Graphiques	Haute	1		

Tableau 1 : : Backlog Produit de notre application Sprint 1

User Stories	Epic	Priorité	Sprint	Début Sprint	Fin Sprint
Création de la base de données	Backend de la gestion des clients	Moyenne	2	25/05/2021	15/06/2021
Gestion du filtre des clients	Backend de la gestion des clients	Moyenne	2		
Consultation des fiches clients enregistrées dans la base de données	Backend de la gestion des clients	Moyenne	2		
Ajout de clients dans la base de données	Backend de la gestion des clients	Moyenne	2		
Modification, correction et suppression de fiches clients	Backend de la gestion des clients	Moyenne	2		

Tableau 2 : : Backlog Produit de notre application Sprint 2

User Stories	Epic	Priorité	Sprint	Début Sprint	Fin Sprint
Backend paramétrage des tables et dossiers fonctionnels	Backend de l'espace de paramétrage	Moyenne	3	16/06/2021	09/07/2021
Backend gestion des listes de valeurs	Backend de l'espace de paramétrage	Basse	3		
Backend Paramétrage du serveur SMTP	Backend de l'espace de paramétrage	Basse	3		
Backend de l'envoi de mails	Backend de l'espace de paramétrage	Basse	3		
Backend du tableau de bord et de l'agenda	Backend de l'espace de paramétrage	Basse	3		

Tableau 3 : : Backlog Produit de notre application Sprint 3

Conclusion : Dans ce chapitre nous avons fait l'analyse et la spécification des besoins du projet en identifiant d'abord les besoins fonctionnels et non fonctionnels. Ensuite nous les avons illustrés à travers une bonne modélisation en utilisant le langage UML. Nous avons fini par présenter le backlog Product qui montre la feuille de route du développement de l'application en ordonnant les fonctionnalités par priorité. Dans le prochain chapitre nous allons passer à la conception détaillée de notre solution.

CHAPITRE 3 : CONCEPTION DE LA SOLUTION

Introduction : Dans ce chapitre, nous allons faire la conception de notre solution. Tout d'abord il s'agira de faire une description des outils et technologies utilisés pour mettre en place la solution. Puis de présenter l'architecture adoptée pour notre application.

I. Outils et technologies

Pour la réalisation de notre application CRM, nous avons utilisé le Framework MIKADO qui est conçu à partir d'outils et technologies tels que J2EE, TOMCAT, XML, ORACLE, HTML, CSS, JavaScript et d'autres technologies que nous avons ajoutées à savoir Bootstrap et JQuery pour gérer les interfaces graphiques.

a. J2EE [6]

J2EE est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

- Un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'applications.
- Un ensemble d'API qui peuvent être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

Sun propose une implémentation minimale des spécifications de J2EE : le J2EE SDK. Cette implémentation permet de développer des applications respectant les spécifications mais n'est pas prévue pour être utilisée dans un environnement de production. Ces spécifications doivent être respectées par les outils développés par des éditeurs tiers.

L'utilisation de J2EE pour développer et exécuter une application offre plusieurs avantages :

- Une architecture d'applications basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement
- La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API
- La possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

b. Le langage JAVA

Java, qui est à la base de J2EE, est un langage de programmation orienté objet. Avant de traiter plus en détails les propriétés de Java coté serveur avec J2EE, il est important de noter que Java est une technologie très présente sur le poste client ainsi que sur les périphériques tels que les téléphones portables, les voitures, les routeurs, les cartes à puces ou encore les assistants numériques. Aussi Java présente de nombreux avantages dont :

- La robustesse
- La sécurité

La première tâche de l'interpréteur Java est de vérifier la conformité du programme, d'analyser si le code n'a pas été transformé entre la compilation et l'exécution du code. Un "byte code vérifié" se charge d'observer si une classe n'a pas accès aux registres, ne manipule pas le « stack » ou n'accède pas au système de fichiers de manière anormale. De plus, Java utilise pour ses transferts une technique de cryptage basée sur le système de clé publique. Le chargement d'une classe Java via le réseau s'opère par le biais d'une « *ClassLoader* » qui gère tous les mécanismes de sécurité. Les applets chargées via le réseau sont restreintes au niveau des accès en lecture et écriture des fichiers locaux, de la création et de la copie de répertoires ou fichiers.

c. Présentation de quelques composants

i. JSP

Les pages JSP sont des composants permettant de générer des pages web dynamiques. Il s'agit de fichiers d'extension « *.jsp* » contenant du html et du Java. Le mécanisme est le suivant : un analyseur lit le code html contenu dans la page ; le code java inscrit dans les balises spécifiques JSP est exécuté au niveau du serveur. Le résultat est du code html ajouté à la page JSP qui sera envoyé au navigateur. Les pages JSP s'exécutent dans un moteur de JSP

ii. Servlets [7]

Une servlet est un programme qui s'exécute côté serveur en tant qu'extension du serveur. Elle reçoit une requête du client, effectue des traitements et renvoie le résultat. La liaison entre la servlet et le client peut être directe ou passer par un intermédiaire comme par exemple un serveur HTTP. Les servlets sont conçues pour agir selon un modèle de requête/réponse. Une servlet peut être invoquée plusieurs fois en même temps pour répondre à plusieurs requêtes simultanées. La servlet se positionne

dans une architecture trois tiers dans le tiers du milieu entre le client léger chargé de l'affichage et la source de données.

d. Serveur d'application : TOMCAT

Tomcat est un serveur open source qui agit comme un conteneur de Servlets J2EE.

Il implémente les spécifications des Servlets et des JSP de Sun Microsystems. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP. Son compilateur jasper compile les pages JSP pour en faire des Servlets et son moteur de Servlets agit souvent en combinaison avec un serveur web (Apache ou autre).

Tomcat a été écrit en langage Java, il peut donc s'exécuter via la JVM (machine virtuelle java) sur l'essentiel des systèmes d'exploitation. Il possède, entre autres, les avantages suivants :

Il est compatible J2EE ; ce qui signifie que les développements réalisés pour Tomcat pourra être portés sur des serveurs d'applications commerciaux comme WebSphere, WebLogic, Oracle Application Server ;

Les développements se font en Java ; ce qui assure un niveau élevé de portabilité : il sera possible de déployer le code qui tourne sur Tomcat sur la plupart des systèmes d'exploitation, (Linux, AS400, Solaris, Windows, etc.). Ce haut niveau de portabilité favorise la réutilisation, atout des technologies objets.

e. Bootstrap [7]

Bootstrap est un Framework CSS. Un Framework correspond à un ensemble de bibliothèques regroupées dans un but précis et possédant des règles internes que doivent suivre les utilisateurs.

En d'autres termes, et pour le dire très simplement, Bootstrap est un ensemble de fichiers CSS et JavaScript fonctionnant ensemble et qu'on va pouvoir utiliser pour créer des design complexes de manière relativement simple.

Le Framework Bootstrap est donc un ensemble de fichiers CSS et JavaScript qui contiennent des règles prédéfinies et qui définissent des composants. Ces ensembles de règles sont enfermés dans des classes et nous n'aurons donc qu'à utiliser les classes qui nous intéressent afin d'appliquer un ensemble de styles à tel ou tel élément HTML.

De plus, Bootstrap utilise également des fichiers JavaScript et notamment des bibliothèques JavaScript externes comme jQuery ou Popper pour définir des composants entiers comme des barres de navigation, des fenêtres modales, etc. qu'on va pouvoir également directement implémenter.

f. JQuery [8]

JQuery est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML. La première version est lancée en janvier 2006 par John Resig (un petit génie du JavaScript).

La bibliothèque contient notamment les fonctionnalités suivantes :

- Parcours et modifications du DOM ;
- Gestion des événements ;
- Effets visuels et animations ;
- Manipulation des CSS ;
- Ajax ;
- Plugins ;
- Utilitaires

II. Architecture de la solution

L'architecture de notre application va s'inspirer du modèle MVC. Dans le modèle MVC, le code est séparé en trois parties :

- **Modèle** : cette partie gère les données de notre application. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. Au niveau du Framework MIKADO, cette couche est représentée par le dossier fonctionnel que nous allons intégrer dans notre application. Ce dernier est une vue logique qui présente une source de données composée d'une ou plusieurs tables liées entre elles avec la description des règles et contraintes sur les données.
- **Vue** : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des informations et de les afficher. Etant donné que le Framework MIKADO ne permet pas la génération de pages dynamiques, nous allons utiliser les technologies Bootstrap, html, CSS, JQuery pour créer nos vues.
- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est le chef d'orchestre : il va demander au modèle les données, les analyser, prendre des décisions et renvoyer les informations à afficher à la vue. Dans notre application, la couche contrôleur est représentée par le module de gestion des fonctionnalités applicatives.

Nous vous présentons à la *figure 21*, la représentation de l'architecture de notre application.

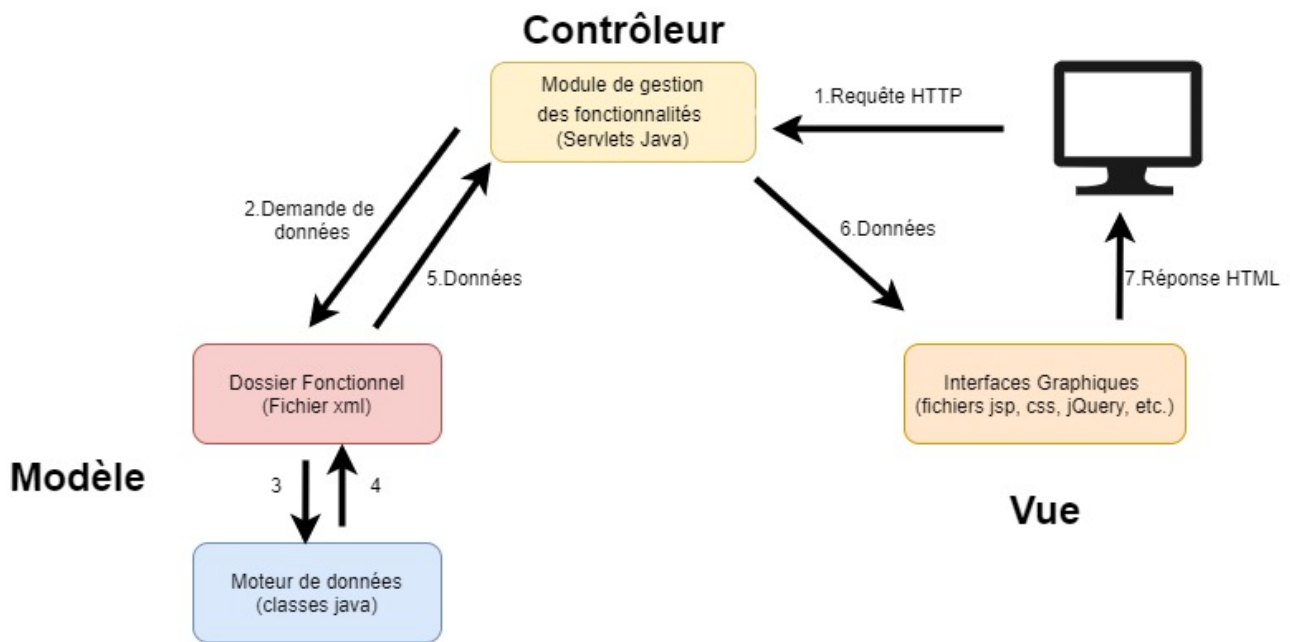


Figure 21 : Architecture de notre application

Conclusion : Ce chapitre consacré à la conception de la solution nous a permis de présenter les composants utilisés lors de l'implémentation et leurs interactions. Nous avons également présenté les outils et technologies utilisés et l'architecture pour la réalisation de notre application. Cela nous permet alors d'implémenter la solution et de la présenter dans le chapitre suivant.

CHAPITRE 4 : PRESENTATION DE LA SOLUTION

Introduction : Dans ce chapitre, nous présentons la solution à travers quelques captures d'écrans. Après avoir présenté l'écran d'accueil, nous montrons toutes les fonctionnalités relatives à l'espace de paramétrage. Ensuite, nous passons aux fonctionnalités concernant la gestion des clients et leurs fiches.

I. Page d'accueil

La *figure 22* représente la page d'accueil de notre application web. Elle se présente sous forme de tableau de bord et donne une analyse synthétique des informations sur les clients. Nous avons représenté :

- L'évolution des clients dans le temps, les clients actifs, les clients inactifs
- Le suivi des mails à savoir le taux de réception, le taux d'ouverture, le taux de réponse
- Le classement des clients par tranche d'âge, etc.

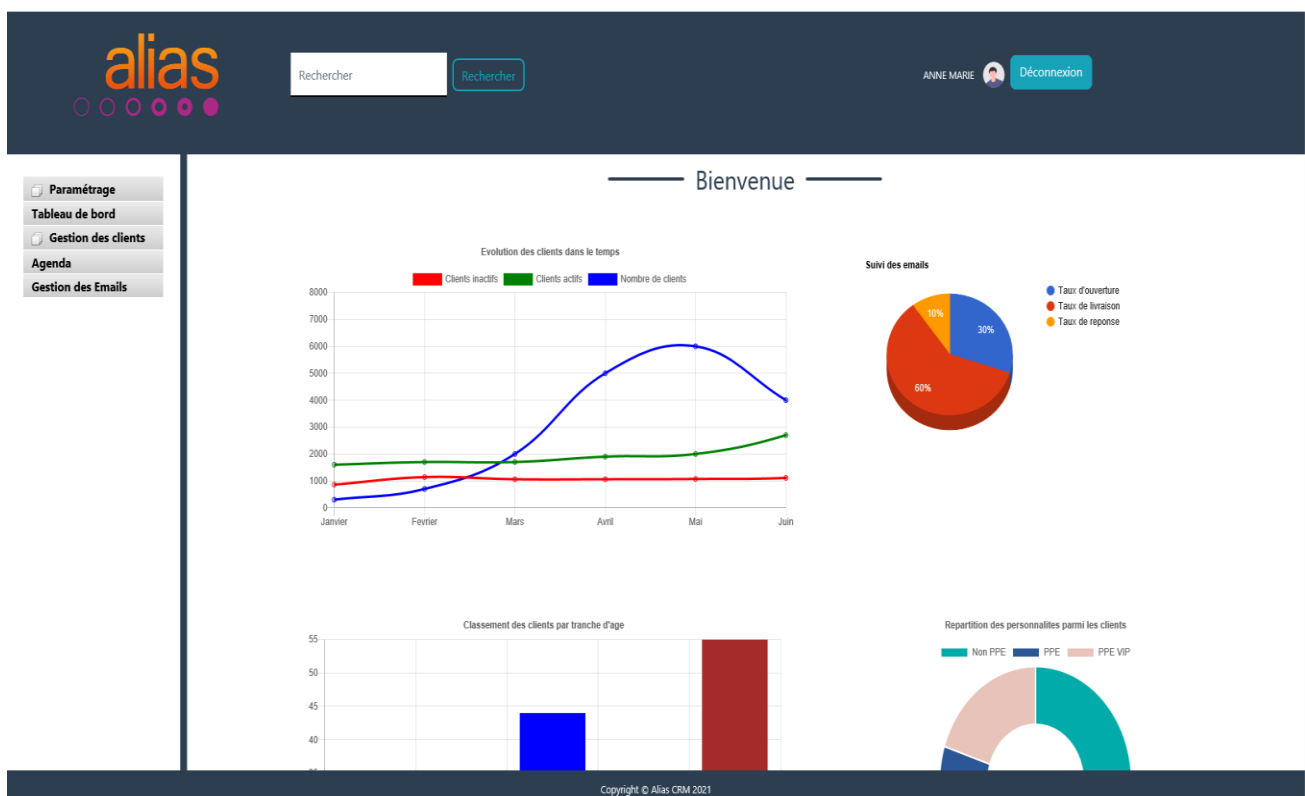


Figure 22 : Page d'accueil

II. Espace de paramétrage

L'espace de paramétrage est réservé à l'administrateur de l'application. Nous pouvons y retrouver la gestion des profils, des utilisateurs, des tables physiques, des dossiers fonctionnels, etc.

a. Gestion des profils

A la *figure 23*, l'administrateur peut créer des profils et les associer à des niveaux d'autorisation. C'est en fonction de ce numéro qu'on détermine quels sont les menus dont il a accès : si notre niveau d'autorisation est inférieur à un numéro d'un menu, on ne peut pas y accéder.

The screenshot shows the 'alias' application interface. The top header has the 'alias' logo and a search bar. The left sidebar contains a 'Paramétrage' (Configuration) menu with sub-items: 'Paramétrage des Utilisateurs de l'application', 'Gestion des Profils de l'application', 'Gestion des Utilisateurs de l'application', 'Paramétrage Listes de valeurs', 'Définition des Tables Physiques', 'Définition des liens MCD', 'Paramétrage Dossier Fonctionnel', and 'Configuration mail-smtp'. Below this are 'Tableau de bord', 'Gestion des clients' (with 'Ajout' and 'Consultation' sub-items), 'Agenda', and 'Gestion des Emails'. The main content area is titled 'Formulaire Profil' and contains an 'Encadré Profil' (Profile Form) with the following fields: 'Code Profil : *' (value: 'resp'), 'Libellé du Profil :' (value: 'Responsable CRM'), and 'Niveau d'autorisation :' (value: '0'). There are 'Enregistrer' (Save) and 'Annuler' (Cancel) buttons at the bottom of the form. A 'Retour à la liste' link is also visible.

Figure 23 : Paramétrage des profils

b. Gestion des utilisateurs

La *figure 24* représente les informations à remplir pour ajouter un nouvel utilisateur. On lui affecte obligatoirement un profil. L'administrateur peut créer autant d'utilisateurs qu'il veut et ces derniers pourront ensuite se connecter pour accéder à l'application.

Paramétrage

Paramétrage des Utilisateurs de l'application

Gestion des Profils de l'application

Gestion des Utilisateurs de l'application

Paramétrage Listes de valeurs

Définition des Tables Physiques

Définition des liens MCD

Paramétrage Dossier Fonctionnel

Configuration mail-smtp

Tableau de bord

Gestion des clients

Agenda

Gestion des Emails

Paramétrage des utilisateurs

Retour à la liste

INFORMATIONS

Titre Utilisateur : MLE - Mademoiselle

Nom : DIOP

Prénom : Anne Marie

Nom Complet Utilisateur : Anne Marie DIOP

Langue utilisateur : FRA - FRANCAIS

Date de Naissance :

Nationalité Utilisateur : SEN - SENEGALAISE

DEFINITION

Login : AMD

Code Service :

Libellé Service :

Lieu de Travail :

Téléphone :

Email :

IDENTIFICATIONS

Code Utilisateur : RES

Méthode de controle : L - LOCAL

Mot de Passe :

Confirmation :

Code Profil : resp

Unité Gestion Production :

DATES

Date d'effet : 06/07/2021

Date de Fin d'effet :

Figure 24 : Ajout d'un utilisateur

c. Gestion des tables physiques

Paramétrage

Paramétrage des Utilisateurs de l'application

Gestion des Profils de l'application

Gestion des Utilisateurs de l'application

Paramétrage Listes de valeurs

Définition des Tables Physiques

Définition des liens MCD

Paramétrage Dossier Fonctionnel

Configuration mail-smtp

Tableau de bord

Gestion des clients

Agenda

Gestion des Emails

Détail table physique

Retour à la liste

Paramétrage Fichier Physique

Nom :

Libellé :

Type :

Nature :

Fichier Xml de Paramétrage :

Emplacement :

Figure 25 : Ajout d'une nouvelle table

A la *figure 25*, nous ajoutons une nouvelle table dans la base de données. Ensuite, lorsqu'on clique sur « paramétrage table », on accède à l'interface représentée par la *figure 26*.

Nous pouvons maintenant ajouter les champs de la table en donnant leurs noms, leurs libellés, leurs types, etc.

Figure 26 : Paramétrage des champs d'une table

d. Configuration du serveur SMTP

L'administrateur doit d'abord paramétrer le serveur SMTP pour permettre le traitement et le transfert des emails. Cela se fait grâce à l'interface montrée à la *figure 27*.

The screenshot displays the Alias application's configuration interface. At the top, the 'alias' logo is on the left, and a search bar with the text 'Rechercher' is on the right. A sidebar on the left lists various configuration options under the 'Paramétrage' heading, including 'Paramétrage des Utilisateurs de l'application', 'Gestion des Profils de l'application', 'Gestion des Utilisateurs de l'application', 'Paramétrage Listes de valeurs', 'Définition des Tables Physiques', 'Définition des liens MCD', 'Paramétrage Dossier Fonctionnel', and 'Configuration mail-smtp'. Below these are links for 'Tableau de bord', 'Gestion des clients', 'Agenda', and 'Gestion des Emails'. The main area shows the 'Configuration' section with a sub-section for 'Configuration SMTP'. This section contains five input fields: 'Serveur SMTP' (with a red asterisk), 'Port', 'Sender par défaut', 'Recevoir par défaut des mails', and 'Password (Facultatif)'. At the bottom right of this section are two buttons: 'Enregistrer' (with a green checkmark icon) and 'Annuler' (with a red X icon).

Figure 27 : Configuration du serveur SMTP

II. Gestion des clients

Les utilisateurs enregistrés peuvent se connecter et gérer les clients en fonction de leur profil. Ils ont la possibilité d'ajouter, de consulter, de modifier les informations des clients, de faire le suivi de façon globale.

a. Ajout d'un client

La *figure 28* est une interface à travers laquelle on peut ajouter un client. Elle montre les différentes fiches à remplir sous forme d'onglets.

alias

Rechercher

ANNE MARIE Déconnexion

Paramétrage

- Paramétrage des Utilisateurs de l'application
- Gestion des Profils de l'application
- Gestion des Utilisateurs de l'application
- Paramétrage Listes de valeurs
- Définition des Tables Physiques
- Définition des liens MCD
- Paramétrage Dossier Fonctionnel
- Configuration mail-smt

Tableau de bord

Gestion des clients

- Ajout
- Consultation

Agenda

Gestion des Emails

Client Etat Civil Adresse Pièces Administratives Contact Informations fiscales Informations supplémentaires Invalidité Patrimoine

PPE VIP Revenu Santé

Identifiant des infos fiscales

Quotient Familial

N° de l'identité

Code Fiscal

Date Effet des Informations

Situation Familiale Fiscale

Date de fin d'effet des Informations

Exercice fiscal de référence (MMAAAA)

Régime Matrimonial

Impot Société

Nombre d'Enfants

Date Exercice Fiscale

Nombre d'Enfants à charge

Citoyen américain

Imposition

Résident américain

Figure 28 : Ajout d'un client

b. Affichage Post IT d'un client spécifique grâce au filtre

A la figure 29, l'utilisateur commence par entrer le numéro d'un client dans le filtre pour consulter ses informations. Il est également prévu d'ajouter d'autres filtres étant donné qu'on peut ne pas retenir le numéro d'un client.

alias

Rechercher

ANNE MARIE Déconnexion

Paramétrage

- Paramétrage des Utilisateurs de l'application
- Gestion des Profils de l'application
- Gestion des Utilisateurs de l'application
- Paramétrage Listes de valeurs
- Définition des Tables Physiques
- Définition des liens MCD
- Paramétrage Dossier Fonctionnel
- Configuration mail-smt

Tableau de bord

Gestion des clients

- Ajout
- Consultation

Agenda

Post It Liste

Client

N° de l'identité

Type de l'identité

Date Effet des informations

Code Adresse

Pièces Administratives +

Contact +

Informations fiscales +

Figure 29 : Filtre pour les clients

Le système nous renvoie ensuite les informations du client sélectionné comme sur la figure suivante.

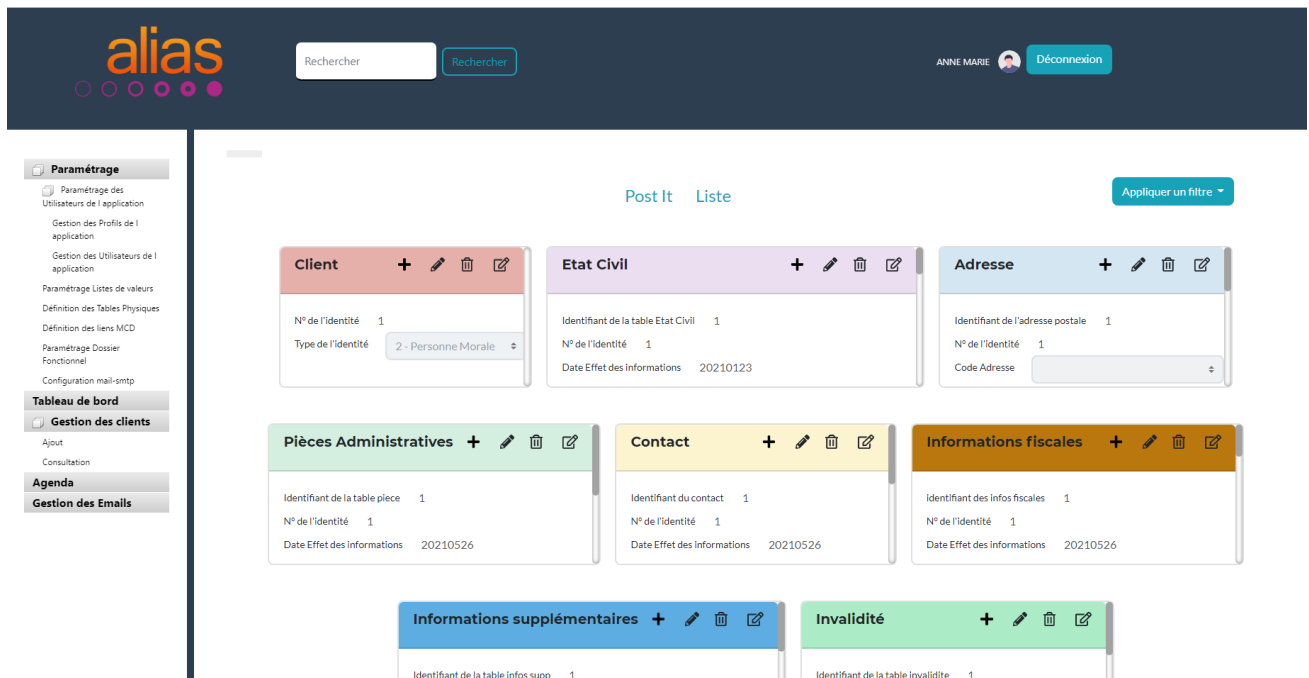


Figure 30 : Consultation des informations d'un client

c. Affichage sous forme d'onglets d'un client spécifique grâce au filtre

L'utilisateur a un deuxième choix quant à l'affichage des informations d'un client : c'est l'affichage sous forme d'onglets.

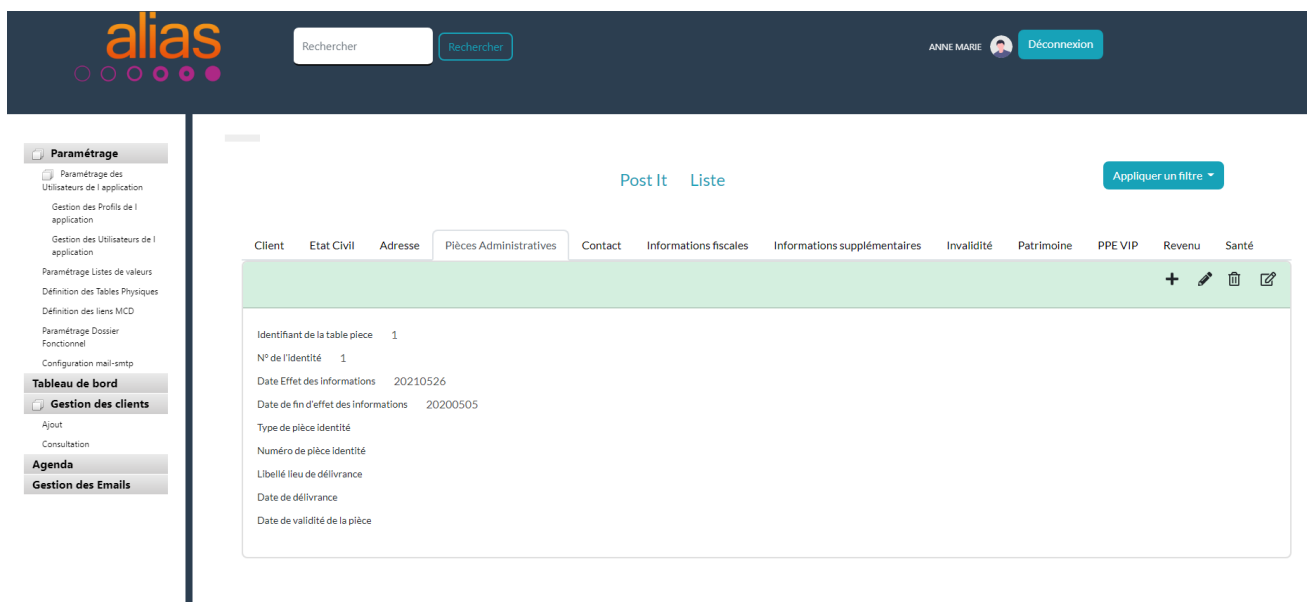


Figure 31 : Affichage des informations d'un client sous forme d'onglets

d. Gestion de l'agenda

L'utilisateur a la possibilité de gérer son agenda à travers notre application CRM. Il peut ajouter ou supprimer des événements et avoir de petits rappels de temps en temps.

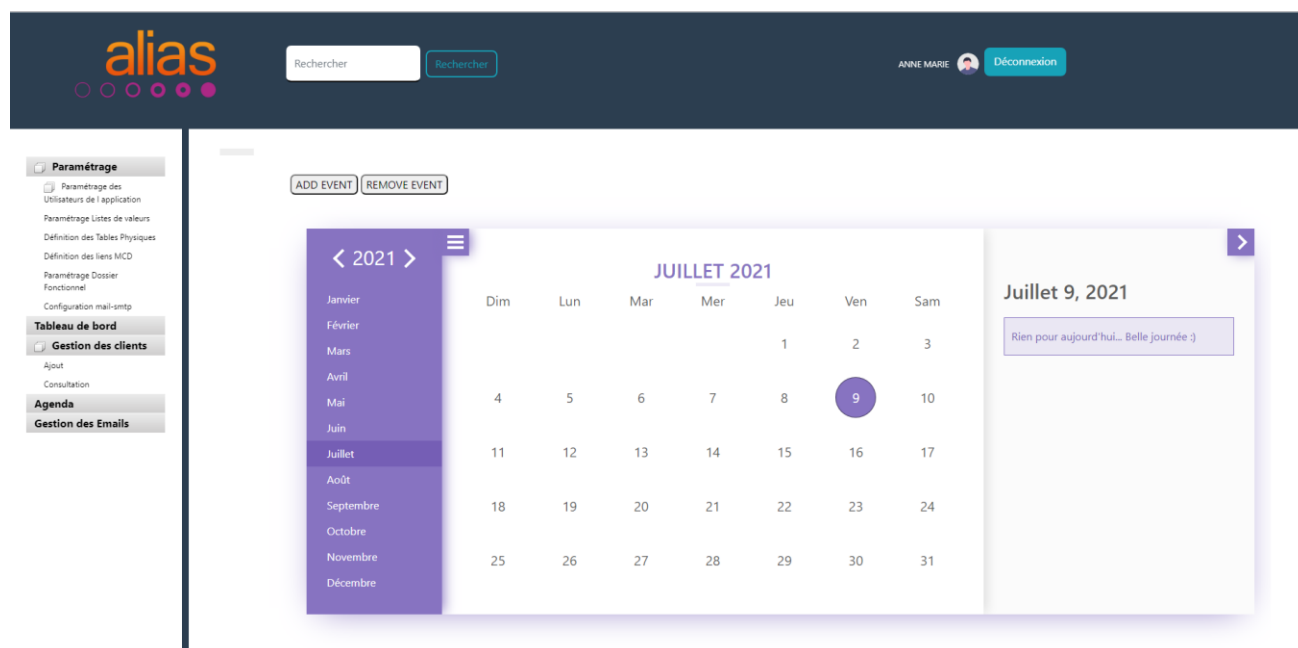


Figure 32 : Gestion de l'agenda

e. Envoi de mails

Après avoir paramétré le serveur SMTP, l'utilisateur peut maintenant envoyer des mails à ses clients à travers une interface comme sur la figure 33.

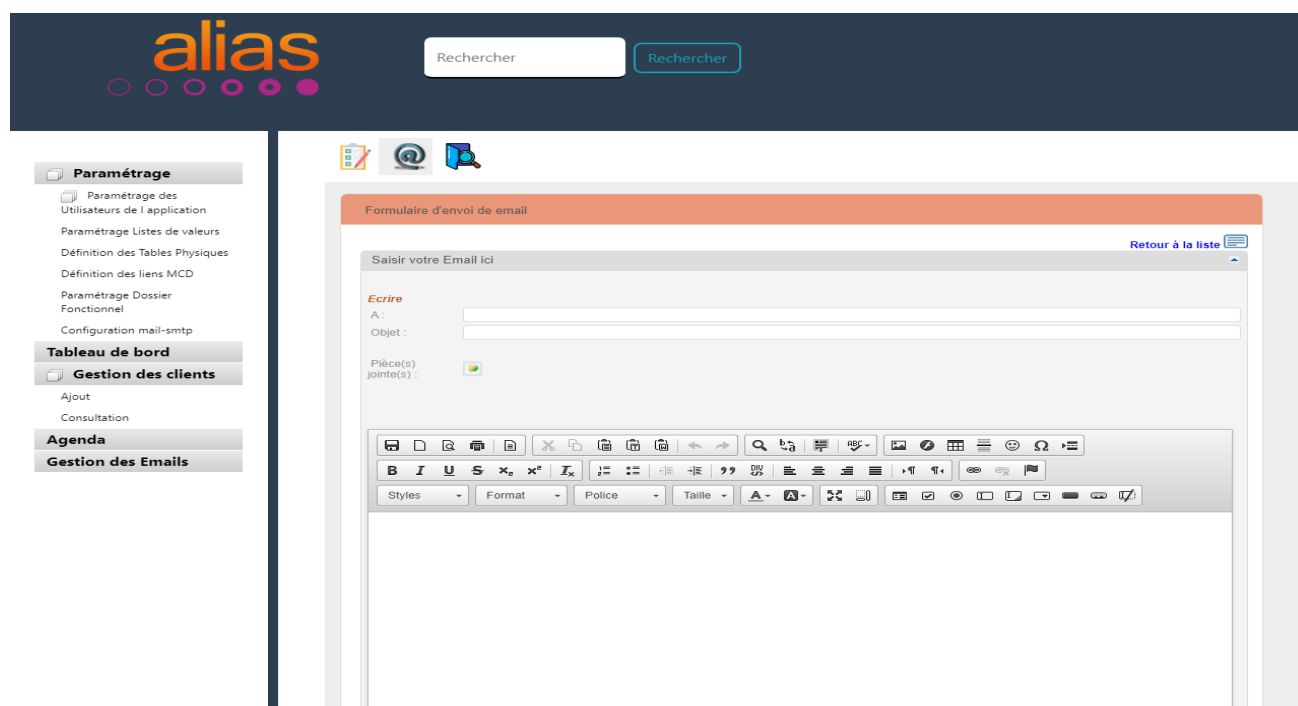


Figure 33 : Envoi de mails

Conclusion : Dans ce dernier chapitre, nous avons montré la solution mise en œuvre à travers des captures d'écrans. Ces derniers donnent un aperçu de l'ERP orienté CRM que nous avons développé.

Conclusion

Durant notre stage effectué au sein de l'entreprise SOPartech Développement, nous avons travaillé sur la mise en place d'un ERP orienté CRM pour la gestion de la relation client. L'objectif de ce projet a été de mettre en place une application qui permet de gérer les relations et interactions avec chaque client d'une entreprise et la possibilité de personnaliser l'application à travers un espace de paramétrage. Nous avons pu développer la partie paramétrage des tables de la base de données, des champs et leurs types de façon dynamique, des liens entre les tables, des utilisateurs et leurs profils. Les utilisateurs créés peuvent à leur tour gérer les clients, leur agenda, l'envoi de mails et avoir accès à un tableau de bord. L'essentiel de l'espace de paramétrage a été réalisé ainsi que le module de gestion des clients.

Pour atteindre ces différents objectifs, nous avons, par ailleurs, adopté la méthodologie agile Scrum tout au long du projet et le langage UML pour une représentation de notre système sous formes de diagrammes d'analyse. Comme outils et technologies utilisés pour la réalisation de notre système, nous pouvons citer le Framework MIKADO, J2EE, Bootstrap, JQuery, HTML, CSS, Javascript, entre autres.

Ce stage nous a offert l'opportunité de nous imprégner du monde de l'entreprise et d'approfondir nos connaissances. Il nous a aussi permis de monter en compétence sur des technologies et outils de pointe qui enrichissent nos connaissances théoriques acquises au cours de ces cinq années d'études.

En perspective, nous envisageons d'abord de terminer les fonctionnalités de modification et de correction de fiches clients, ensuite nous comptons gérer les fonctionnalités de déplacement (drag and drop) des fiches clients affichées sous format post IT et enfin rendre paramétrable le tableau de bord et le menu.

Références

- [1] Extel. [En ligne]. Available: <https://www.extel.fr/index.php>. [Accès le 03 2021].
- [2] «Qu'est-ce qu'un ERP?», [En ligne]. Available: <https://www.cegid.com/fr/faq/quest-quun-erp/>. [Accès le 07 2021].
- [3] «qu'est-ce-qu'un CRM?», [En ligne]. Available: <https://www.salesforce.com/fr/learning-centre/crm/what-is-crm/>.
- [4] D. Galiana, «7 méthodologies de gestion de projet à connaître absolument», [En ligne]. Available: <https://www.planzone.fr/blog/methodologies-gestion-projet>. [Accès le 04 2021].
- [5] D. Galiana, «Qu'est-ce que la méthodologie Scrum», 2017. [En ligne]. Available: <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-scrum>.
- [6] «Presentation d'UML», [En ligne]. Available: https://www.memoireonline.com/04/15/9079/m_Conception-d-un-site-de-vente-des-produits-dans-un-etablissement-commercial-cas-des-etablissement8.html. [Accès le 04 2021].
- [7] «Développons en Java», [En ligne]. Available: <https://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>.
- [8] J.-M. DOUDOUX, «Developpons en java», [En ligne]. Available: <https://www.jmdoudoux.fr/java/dej/chap-servlets.htm>. [Accès le 04 2021].
- [9] P. Giraud, «Présentation de Bootstrap», [En ligne]. Available: <https://www.pierre-giraud.com/bootstrap-apprendre-cours/introduction/>.
- [10] «Présentation de jQuery», [En ligne]. Available: https://sutterlity.gitbooks.io/apprendre-jquery/content/presentation_de_jquery.html.

Résumé

Ce document présente la réalisation d'un projet effectué au sein de l'entreprise SOPartech développement et s'insère dans le cadre d'un stage de fin de cycle pour l'obtention du Diplôme d'Ingénieur de Conception en Informatique. Notre travail est intitulé : Création d'un ERP orienté CRM pour la gestion de la relation.

Présenté en quatre chapitres, le projet décrit dans ce document a pour objectif de mettre en place une application qui gère les informations des clients d'une entreprise tout en donnant à cette dernière la possibilité de paramétrer l'application.

Nous avons commencé par étudier le système, recenser les différents besoins et faire une bonne modélisation. Ensuite, nous avons mis en place l'interface de paramétrage de l'application, l'interface de gestion des clients et les traitements associés en nous assurant de remplir les critères d'ergonomie. La solution obtenue est présentée avec quelques captures d'écran.

Mots clés : CRM, ERP relation client, MIKADO, paramétrable, fiches client, SCRUM, UML.

Abstract

This document presents the realization of a project carried out within the company SOPartech développement and is part of an end-of-cycle internship to obtain the Diploma of Computer Design Engineer. Our work is titled: Creation of a CRM-oriented ERP for customer relationship management.

Presented in four chapters, the project described in this document aims to set up an application that manages the information of a company's customers while giving the possibility to parameterize the application.

We started by studying the system, identifying the different needs and doing a good modelling. Then, we implemented the application's setting interface, the customer management interface and the associated processes, making sure to provide the ergonomic criteria. The resulting solution is presented with some screenshots.

Keywords: CRM, ERP customer relationship, MIKADO, customizable, customer files, SCRUM, UML.