



Ministère de l'Enseignement Supérieur,
De la recherche Scientifique et de la Technologie
Université du Sousse



UNIVERSITE DE SOUSSE
INSTITUT SUPERIEUR DE GESTION DE SOUSSE

Rapport de Projet de Fin d'Etudes

Présenté en vue de l'obtention

Du diplôme de la Licence Fondamentale en Informatique de Gestion

SUJET :

Conception et réalisation d'une application Web

PINGOUIN MAILING

Préparé sous la direction de :

- ❖ Mr BESSAAD Oussama
- ❖ Mr SOUSSI Haithem

Présenté et soutenu publiquement par :

- ❖ BELLAZREG Anis

Remerciement

C'est un grand plaisir que nous réservons cette page en signe de notre sincère gratitude envers tous ceux qui nous ont aidés de près ou de loin au bon déroulement de ce travail.

Nous tenons à remercier spécialement notre encadreur interne Mr BESSAAD Oussama qui avec un grand dévouement, a consacré beaucoup de temps à suivre de près l'évolution de notre projet et nous a réservé les conditions saines de travail, pour ses directives constructives et l'aide précieuse qu'il nous a apportée.

Nous avons le plaisir d'exprimer nos vifs remerciements aux personnels de la société Pingouin, pour leur hospitalité et à notre encadreur Mr SOUSSI Haithem qui avec un grand dévouement, a consacré beaucoup de temps à suivre de près l'évolution de notre projet et nous réserve les conditions saines de travail, pour ses directives constructives à l'aide précieuse qu'il nous a apporté.

A notre chef d'appartement :

OMRANI Walid

Pour son dévouement et son extrême gentillesse.

Aux membres de jury

&L

A tous nos enseignants :

Qui jugeront ce projet, ou ils y trouveront l'expression de nos grandes considérations et nos grands respects.

Dédicaces

Je dédie ce travail à tous ceux que j'aime.

À dieu le tout puissant :

À dieu qui m'a donné l'effort et le courage, pour la réalisation de ce travail.

À mon cher père « Ameur »

Qui est le meilleur père dans ce monde, qui n'a jamais cessé de me fournir son assistance et son amour infini, il m'a appris le vrai sens de la vie, du respect et de générosité. Grace à son encouragement, ses sacrifices qu'il a consenti pour mon succès, sa confiance et son soutien moral et matériel et en exprimant mes grâitudes, mon profond amour et ma passion.

À ma chère mère « Najwa »

Qui a veillé sur moi et a tout souffert sans me faire souffrir. Pour votre amour, votre patience et votre générosité, pour tous les efforts que vous avez consentis en ma faveur. J'espère avoir été digne de votre affection et de votre confiance.

Que dieu vous accorde bonheur, bonne santé et longue vie

À mes cher frères « Mahdi et Riadh »

Merci pour vos encouragements et vos conseils. Je vous souhaite une vie pleine de succès, de joie et que Dieu nous garde unis.

À mes chers amis « Badri, Radhwane, Ahmed, Siwar, Amani, et Rim.

En souvenir de ces années que nous avons passées ensemble. Vous restez dans mes pensées et dans mon cœur.

Anis

Table des matières

Introduction

Chapitre I : Présentation de stage

I.	Introduction :	1
II.	Présentation générale de « La Société Pingouin » :	1
1.	Organisation :	1
2.	Adresse et coordonnées :	2
III.	L'équipe de la société(Pingouin) :	2
IV.	Les services:	3
V.	Conclusion :	5

Chapitre II : Etude préalable

I.	Introduction :	6
II.	Etude de l'existant :	6
III.	Critique de l'existant :	6
IV.	Solutions à prévoir :	7
V.	Objectifs :	7
VI.	Spécification des besoins :	8
1.	Besoins fonctionnels :	8
2.	Besoin non fonctionnel :	9
VII.	Présentation du notre application:	9
1.	Introduction :	9
2.	Charte graphique et charte éditoriale :	10
3.	Prototype des interfaces de l' administrateur :	11
4.	Prototype des interfaces client :	12
VIII.	Conclusion :	13

Chapitre III: Etude Conceptuelle

I.	Introduction :	14
II.	Architecture du système :	14
III.	Choix du langage de conception :	18
1.	Le choix d'UML :	18
2.	Les points forts d'UML :	20
3.	Les points faibles d'UML :	21
IV.	Conception :	21

1.	Diagramme de Cas d'utilisation :	21
1.1	Définition :	21
1.2	Concepts de base :	22
1.3	Diagramme de cas d'utilisation initiale :	23
1.3.1.	Raffinement de cas d'utilisation « Gestion de compte utilisateur » :	24
1.3.2.	Raffinement de cas d'utilisation « Authentification » :	25
1.3.3.	Raffinement de cas d'utilisation « Gestion des messages » :	26
1.3.4.	Raffinement de cas d'utilisation « Gestion des contacts » :	27
1.3.5.	Raffinement de cas d'utilisation « Gestion des listes »:	27
1.3.6.	Raffinement de cas d'utilisation « Gestion de calendrier »:	28
1.3.7.	Raffinement de cas d'utilisation « Gestion messages contact »:	29
1.4	Cas d'utilisation Finale :	30
2.	Diagramme de séquence :	31
2.1	Définition de diagramme de séquence :	31
2.2	Diagramme de séquence « Inscription » :	31
2.3	Diagramme de séquence « Authentification » :	32
2.4	Diagramme de séquence de client « Composer message »:	33
2.5	Diagramme de séquence de client « Envoyer message » :	34
2.6	Diagramme de séquence de client « Ajouter contact » :	35
2.7	Diagramme de séquence de client « Modifier contact » :	36
2.8	Diagramme de séquence de client « Importer contacts » :	37
2.9	Diagramme de séquence de client « modifier liste des contacts » :	38
2.10	Diagramme de séquence d'administrateur « Ajouter client » :	39
2.11	Diagramme de séquence d'administrateur « Supprimer client » :	40
2.12	Diagramme de séquence d'administrateur « Bloquer client » :	41
2.13	Diagramme de séquence « Ajouter événement au calendrier » :	42
2.14	Diagramme de séquence de client « Envoyer message contact »:	43
3.	Diagramme de collaboration :	44
3.1	Diagrammes de collaboration « Ajouter contact».....	44
3.2	Diagrammes de collaboration « Modifier contact» :	45
3.3	Diagrammes de collaboration « Supprimer contact»	45
4.	Diagramme d'activité :	46
4.1	Diagramme d'activité « Authentification » :	46
4.2	Diagramme d'activité « Ajouter contact » :	48
4.3	Diagramme d'activité « Modifier contact » :	49
4.4	Diagramme d'activité « supprimer contact » :	50
5.	Diagramme de déploiement :	51
6.	Diagramme de classe :	52

Chapitre IV : Réalisation

I.	Introduction	54
II.	Langages et outils de développement :.....	54
1.	HTML :	54
2.	CSS :.....	54
3.	Adobe photoshop CS5 :.....	55
4.	Adobe Dreamweaver CS5 :.....	56
5.	PHP (Personal Home Page) :	57
5.1	Définition :	57
5.2	Fonctionnement de PHP :.....	58
5.3	Avantages et Limitations de langage PHP:	58
6.	WampServer:.....	59
6.1	PHPMyAdmin :.....	60
6.2	Apache :.....	60
6.3	MYSQL :.....	60
7.	JQuery	61
8.	Power AMC :	62
9.	Rational Rose :	62
III.	Réalisation :.....	63
1.	« Planning » de réalisation :	63
2.	Etape réalisation du système :	64
3.	Les interfaces coté client :.....	65
3.1	Interface d'inscription:	65
3.2	Interface d'authentification :	66
3.3	Interface gérer listes des contacts :.....	67
3.4	Interface gérer les contacts :.....	68
3.5	Interface importer/exporter des clients:.....	68
3.6	Interface composer messages:	69
3.7	Interface envoyer messages: (dynamique)	69
3.8	Interface envoi automatique des messages: (dynamique)	70
3.9	Interface gérer messages:	70
3.10	Interface gestion de calendrier :	71
3.11	Interface statistique d'envoi:.....	72
3.12	Interface menu administrateur : (page statique).....	73
3.13	Interface gestion messages contact	74
3.14	Interface statistique de Google Analytics :	75

Conclusion

Liste des figures

Figure 1 : Organisation de la société	2
Figure 2 : Organigramme de la société	3
Figure 3 : L'équipe de la société	3
Figure 4 : Design et Expérience utilisateur	4
Figure 5 : Hébergement web	5
Figure 6 : Architecture du système	15
Figure 7 : Modèle du cycle de vie en cascade	17
Figure 8: UML	18
Figure 9 : Vue Statique.....	19
Figure 10 : Vue Dynamique	20
Figure 10 : Diagramme de cas d'utilisation initial	23
Figure 12 : Diagramme de cas d'utilisation « Gestion de compte utilisateur »	24
Figure 13 : Diagramme de cas d'utilisation « Authentification »	25
Figure 14 : Diagramme de cas d'utilisation « Gestion des messages »	26
Figure 15 : Diagramme de cas d'utilisation « Gestion des contacts »	27
Figure 16 : Diagramme cas d'utilisation « Gestion des listes »	27
Figure 17 : Diagramme cas d'utilisation « Gestion de calendrier ».....	28
Figure 18 : Diagramme cas d'utilisation « Gestion de messages contact »	29
Figure 19 : Diagramme cas d'utilisation « Gestion de messages contact »	30
Figure 20 : Diagramme de séquence « Inscription ».....	31
Figure 21 : Diagramme de séquence « Authentification ».....	32
Figure 22 : Diagramme de séquence « Composer message »	33
Figure 23 : Diagramme de séquence « Envoyer message »	34
Figure 24 : Diagramme de séquence « Ajouter contact »	35
Figure 25 : Diagramme de séquence « Modifier contact »	36
Figure 26 : Diagramme de séquence « Importer contacts »	37
Figure 27 : Diagramme de séquence « Modifier liste des contacts ».....	38
Figure 28 : Diagramme de séquence d'administrateur « Ajouter client ».....	39
Figure 29 : Diagramme de séquence d'administrateur « Supprimer client ».....	40
Figure 30 : Diagramme de séquence d'administrateurs « Bloquer client »	41
Figure 31 : Diagramme de séquence de client « Ajouter événement au calendrier »	42
Figure 32 : Diagramme de séquence de client « Envoyer message contact »	43
Figure 33 : Diagramme de collaboration « Ajouter contact ».....	44
Figure 34 : Diagramme de collaboration « Modifier contact ».....	45
Figure 35 : Diagramme de collaboration « Supprimer contact »	46
Figure 36 : Diagramme d'activité « Authentification »	47
Figure 37 : Diagramme d'activité « Ajouter contact ».....	48
Figure 38 : Diagramme d'activité « Modifier contact ».....	49
Figure 39 : Diagramme d'activité « Supprimer contact »	50
Figure 40 : Diagramme de déploiement	51
Figure 41 : Diagramme de classe	52
Figure 42 : HTML	54
Figure 43 : CSS	54
Figure 44 : Adobe photoshop CS5	55
Figure 45 : Adobe Dreamweaver CS5	56
Figure 46 : PHP	57
Figure 47 : L'interprétation PHP génère du code HTML	58
Figure 48 : WampServer	59

Figure 49 : PHPMyAdmin	60
Figure 50 : Apache	60
Figure 51 : MYSQL	61
Figure 52 : JQuery	61
Figure 53 : Power AMC	62
Figure 54 : Rational Rose	62
Figure 55 : Interface d'accueil.....	64
Figure 56 : Interface d'inscription.....	65
Figure 57 : Interface d'authentification utilisateur.....	66
Figure 58 : Interface Menu principale de client	67
Figure 59 : Interface Gérer les contacts.....	68
Figure 60 : Importation d'un contact.....	68
Figure 61 : composer un message	69
Figure 62 : Envoyer un message	69
Figure 63 : Envoi automatique d'un message	70
Figure 64 : Interface gérer messages.....	70
Figure 65 : Interface calendrier	71
Figure 66 : Interface statistique d'envoi.....	72
Figure 67 : Menu principale de l'administrateur.....	73
Figure 68 : Gestion de messages contact	74
Figure 69 : Interface statistique de Google Analytics	75

Liste des tableaux

Tableau 1 : Prototype des interfaces administrateur	11
Tableau 2 : Prototype des interfaces client.....	12
Tableau 3 : Acteurs de système.....	22
Tableau 4 : Le « Planing » de réalisation	63

Introduction

Dans le contexte actuel des organisations, marqué essentiellement par la multiplicité des échanges à l'échelle mondiale, l'accroissement des rivalités concurrentielles, et la rapidité des innovations, l'entreprise évolue dans un environnement de plus en plus complexe où la maîtrise de l'information est devenue un pilier stratégique dans le circuit du système d'information.

L'architecture des applications modernes d'entreprise est le résultat d'un processus de sélection naturelle rapide où les critères de compétitivité s'appellent modularité, maintenabilité, portabilité, efficacité, évolutivité... Sous la pression de ces critères et du continual renouvellement des technologies, l'architecture des applications a progressivement évolué du système Mainframe à l'application Web en passant par le client / serveur et les systèmes distribués.

Au cœur de ces évolutions, les applications Web a vu le jour. Issue du mariage entre les systèmes d'information et le Web, et regroupe toute les outils ayant pour objectif d'aider les entreprises à bien connaître et gérer les différents entités et services offertes.

L'application mailing qui nous a été proposée par « Pingouin » sera en mesure d'offrir des services de mailing et de créer un historique des messages vis-à-vis des emails envoyés par les clients. Il serait alors possible d'ajouter des nouveaux contacts, de composer des modèles des messages qui permettent aux clients d'envoyer un énormes nombre d'email à une liste des contacts qui contiennent même des milliers de contacts, ainsi d'envoyer automatiquement des messages et aussi contrôler l'envoi.

Ce projet de fin d'étude a pour objectif de présenter les différentes étapes de conception de cette application qu'on a baptisé « Application mailing », ainsi que ses différentes composantes.

I. Introduction :

Notre stage s'effectue au sein la société «Pingouin» Société de Service d'information. Notre tâche tourne autour de l'identification des problèmes qui surgissent dans les différents services et spécialement dans le service commercial (problèmes stratégiques, manque d'une application d'envoie de mailing open source qui s'adapte aux besoins de la société).

II. Présentation générale de « La Société Pingouin » :

1. Organisation :

Pingouin est une agence web en Tunisie fondée en 2010 par un jeune ingénieur national en informatique, ancien ingénieur de développement dans la création et la réalisation de site Internet sur une mesure dans une agence web offshore.

Petite structure, aux profils complémentaires, notre relation client est basée sur l'écoute et le dialogue. Comprendre vos besoins et les attentes d'un secteur d'activité, nous permets de concevoir et mettre en ouvre des solutions personnalisées fiables et prennes.

Conception graphique, gestion de contenus dynamique (CMS) (Joomla, spip, etc....), base de données, extranet, référencement, utilisation des dernières standards et technologiques...

Nous mettons à votre disposition toutes nos compétences et savoir-faire pour vous accompagner dans la réussite de votre projet internet !



2. Adresse et coordonnées :

- ➡ Adresse : Av El Hadj Ali Soua, Imm Panorama, 2ème étage, Appt 5, Ksar Hellal 5070 Tunisie
- ➡ E-mail: contact@ping-ouin.com
- ➡ Tél: (+216) 73 454 032
- ➡ Mobile: (+216) 55 607 648

III. L'équipe de société(Pingouin) :

C'est une société qui travaille dans le domaine web et multimédia sous la gérance de Monsieur :Haithem SOUSSI.

L'équipe de société Pingouin est formé de :

Gérant (fondateur) : Haithem SOUSSI.

Service informatique : Un groupe de développeurs



IV. Les services:

Pingouin vous accompagner dans vos projets de design/refonte de sites ou d'implantation d'applicatifs web à travers une panoplie de services à forte valeur ajoutée.



Pingouin vous accompagne dans vos projets de design/redesign de sites ou d'implémentation d'applicatifs web à travers une panoplie de services à forte valeur ajoutée.

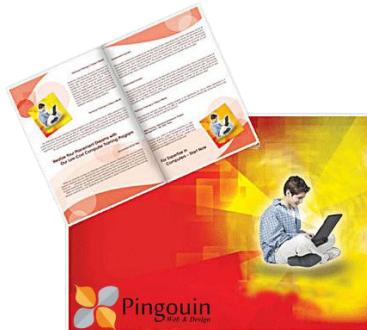
Découvrez nos services couplés à nos solutions de base ou proposés d'une manière séparée en réponse à la problématique précise de nos clients.

Conseil et Stratégie

Le problème pour une PME ou autre n'est pas de créer son site, mais d'adopter une stratégie claire pour attirer les visiteurs et en faire des clients. Nos conseillers sont là pour les aider à déterminer et atteindre ces objectifs !

Design et Expérience utilisateur

Votre site web constitue un excellent moyen de communiquer avec vos publics cibles, surtout si vos visiteurs ont apprécié le « look and feel », trouvé rapidement les informations et vécu une expérience utilisateur agréable.



✚ Webmarketing et SEO

Pingouin vous accompagne pour mettre en œuvre les solutions adéquates pour générer du trafic qualifié en qualité et quantité au meilleur coût et améliorer votre visibilité sur les moteurs de recherche. Voyez comment !



✚ Analytics et Optimisation

La difficulté de l'exercice web, c'est d'atteindre parfaitement sa cible. Pour faire mouche, les entreprises doivent procéder à l'analyse fine de leurs visiteurs et mettre en place un dispositif d'optimisation régulier de leurs sites.



Nom de domaine et Hébergement web

Pingouin vous propose l'enregistrement de nom de domaine, ainsi que l'hébergement web de qualité et tous les services indispensables au bon fonctionnement de votre site Internet.



V. Conclusion :

Dans ce chapitre j'ai présenté le cadre de mon stage à travers une description générale de l'entreprise. Une étude préalable plus détaillée sera réalisée dans le chapitre suivant.

I. Introduction

Dans ce chapitre, nous amenés de développer une application web mailing qui nous permet la description d'un système qui offre des services à l'internaute. À ce point nous posons la problématique avec le présentant de la situation actuel de la société, le critique de l'existant c'est pour la proposition d'une solution.

II. Etude de l'existant

On effectue un stage dans la société Pingouin. Au cours de l'étude du processus de l'envoi des messages à un grand nombre de destinataires, on a pu relever quelques problèmes concernant les services de l'envoi des emails, l'archivage des messages envoyés, import et l'export des contacts et la classification d'un grand nombre de contact par groupe spécifique, de plus les applications mailing qui offrent des services bien avancés ne sont pas gratuit et ils sont trop chère.

III. Critique de l'existant :

On constate :

- ✓ la perte du temps au niveau de l'envoi des messages à chaque destinataire appart
- ✓ redondance
- ✓ Manque au niveau des services offerts à l'internaute
- ✓ Manque d'un archive qui contient les messages déjà envoyés aux listes des contacts
- ✓ Pas de modèle de présentation des donnés des contacts enregistrés à la base de données
- ✓ Manque de service de blocage des contacts en mettant à la liste noir.
- ✓ Manque de service de contrôle des messages par l'administrateur.
- ✓ Manque d'un service qui permet de contrôler l'envoi et la réception des messages par les contacts
- ✓ Manque de service d'envoi automatique des messages contrôlé par l'administrateur.
- ✓ Manque d'un calendrier qui contienne les différents événements intéressant du client.
- ✓ Manque de contact entre l'administrateur et l'utilisateur.
- ✓ Le risque de perte des données des messages et des contacts
- ✓ Problème de sécurité des données des clients contre toute menace
- ✓ Interface client est compliquée surtout pour les nouveaux utilisateurs.

IV. Solution à prévoir :

Pour résoudre ces différents problèmes, nous proposons les solutions suivantes :

- Développer une application mailing pour offrir des services gratuits à la société tournant sur un réseau hétérogène.
- L'envoi des messages à un grand nombre des contacts en une seule fois.
- les droits d'accès sont bien spécifiques en séparant l'espace clients de l'espace administrateur.
- Gestion des contacts en utilisant aussi la fonctionnalité d'import et d'export des contacts pour ne pas perde les données.
- Gestion des messages en créant des modèles prête à envoyés aux contacts à tout moment.
- Gestion d'envoi automatique des messages selon date et heure, et ce service reste contrôlé par l'administrateur.
- Gestion de la base de données avec sécurité surtout aux niveaux des données des clients.
- Gestion de calendrier qui contienne les événements importants par le client.
- Contrôle des messages crées par les clients.
- Gestion de service contact entre l'administrateur et les autres utilisateurs pour améliorer l'application.
- Minimum de temps aux niveaux de l'étape de l'envoi des messages à différentes listes des contacts
- Création des statistiques d'envoie et de réception des messages envoyés aux contacts
- Etat de sortie de l'application c'est la déconnexion en toute sécurité

V. OBJECTIFS :

La mise en place d'un outil permet:

- Envoi de messages à une grande masse de contacts.
- Le Suivi des messages envoyés aux contacts
- Création d'une interface qui repend aux critères ergonomique, qui est simple à utiliser.
- Les Différentes taches gérer par le client :
 - Inscription
 - Authentification
 - Gestion des contacts
 - Gestion des listes des contacts
 - Gestion des messages
 - Envoi des messages
 - Gestion de listes noires

- Suivie des contacts
 - Gestion de calendrier
 - Envoi de message à l'administrateur
- Les Différentes taches gérées par l'administrateur:
- Authentification
 - Gestion des clients
 - Gestion des messages de contacts
 - Gestion des messages créés par les clients
 - Activer/désactiver l'envoi automatique des messages.

VI. Spécification des besoins :

A. Besoins fonctionnels :

Mise en place d'une application Web qui permet l'envoi des messages à une grande masse de contacts. Il doit satisfaire les principaux besoins fonctionnels qui se présentent dans les points suivants :

- L'authentification d'utilisateur est obligatoire pour chaque ouverture de session.
- Protection des données de clients contre toutes menaces.
- Développement d'un système configurable avec Google analytics.
- Contrôle des messages envoyés aux clients.
- Les fonctions de gestion de messages:
- La consultation des messages déjà créés.
 - Créer, Modifier, supprimer des messages
 - Envoi des messages
 - Envoi automatique des messages.
- Les fonctions de gestion de contacts:
- La consultation de la liste des contacts.
 - Ajouter, Modifier, supprimer des contacts
 - Import et export des contacts
- Les fonctions de gestion de listes:
- La consultation de la liste des listes de contacts.
 - Ajouter, Modifier, supprimer des listes
- Les fonctions de gestion de calendrier :

- La consultation de calendrier.
 - Ajouter, Modifier, supprimer les événements de calendrier.
- Les fonctions de gestion des clients (gérer par l'administrateur):
- La consultation de la liste des clients.
 - Ajouter, Modifier, supprimer et bloquer des clients

B. Besoin non fonctionnel

L'application doit répond à tous ces critères :

- Le site doit être performant, fiable et facile à utiliser.
- Les tâches doivent être indépendantes pour ne pas se bloquer à une phase spécifique.
- Le site doit être basé sur une conception orientée objet.
- Le site doit être sécurisé et doit accepter les améliorations.
- Le site est convivial.

VII. Présentation du notre application:

1. Introduction :

Nous souhaitons apporter une solution complète de d'une application mailing qui permet aux clients d'envoyer un email à un énorme nombre de contacts, aussi il permet l'envoi automatique des messages, et il offre d'autres services aux clients de l'application.

2. Charte graphique et charte éditoriale :

La charte graphique est conçue pour être esthétiquement agréable pour les clients et les divers acteurs de l'application, tout en restant ergonomique et intuitive.

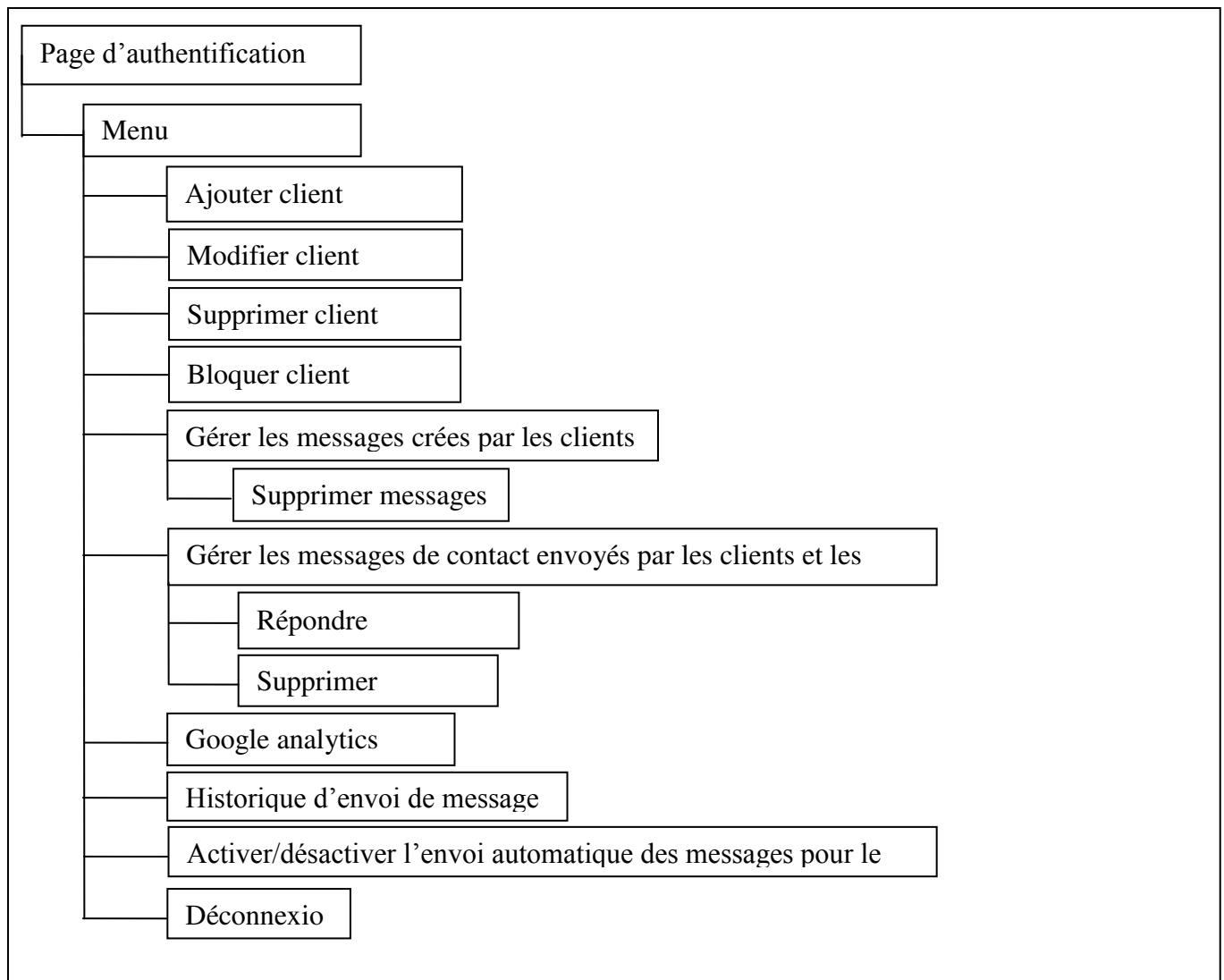
La charte graphique est conforme aux standards et aux normes dictées par le W3C ainsi que les usages du graphisme et du Web.

Cette application web dispose d'une interface graphique web simpliste. C'est-à-dire qu'on demande des pages très simples et minimalistes (il ne s'agit pas de faire du HTML complexe).

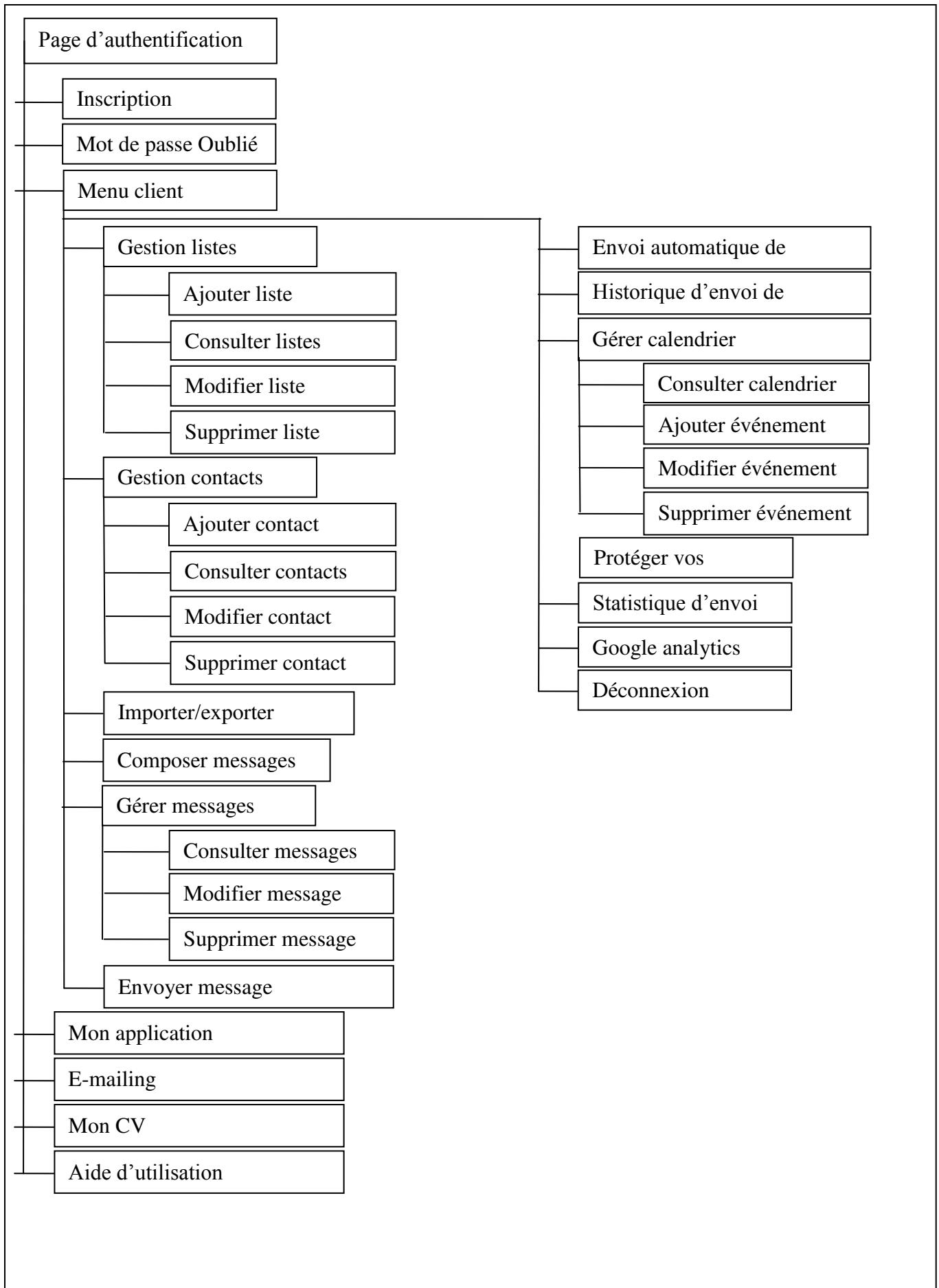
En réalité il existe trois profils de personnes qui peuvent utiliser l'application web de mailing :

- Administrateur de l'application web: il a tous les droits d'accès sur la base de données.
- Client : il bénéficie de la plupart des fonctions car il a les possibilités de gérer les contacts et la gestion des messages.
- Visiteur : il peut contacter l'administrateur.

3. Prototype des interfaces administrateur



4. Prototype des interfaces client



VIII. Conclusion

Dans ce chapitre nous avons spécifié à l'étude de la critique de l'existant, nous avons identifié nos objectives et présenté une étude préalable qui nous a aidés pour comprendre le contexte du notre applications web. Dans le chapitre suivant, on va déterminer les fonctionnalités et les acteurs le plus pertinents.

I. Introduction

Dans ce chapitre nous présentons les méthodologies de conception que nous avons utilisée dans la conception de notre projet

Donc nous présentons une spécification et une analyse des besoins en décrivant les différents outils qui nous avons utilisés dans la conception de notre application web suivit d'une analyse détaillé de besoins explicites et implicites du travail demandé ainsi les différents besoins opérationnel.

Nous enchainons alors avec l'architecture de notre application et par suite nous présentons les différents diagrammes, à savoir les, de cas d'utilisation, de séquence , de classe, d'activité et de collaboration afin de spécifier de façon détaillée les aspects fonctionnels, dynamiques et statiques du système.

II. Architecture du système :

L'objectif de la modélisation constituer de mieux comprendre le fonctionnement du système. Qui est une application mailing

Spécifications ou conception générale

L'architecture d'une application web présente traditionnellement trois niveaux (on parle d'architecture à trois tiers) :

- Tiers client qui correspondent à la machine sur laquelle l'application cliente est exécutée.
- Tiers métiers qui correspondent à la machine sur laquelle l'application centrale est exécutée.
- Tiers accès aux données qui correspondent à la machine gérant la stockage des données.

Chaque tiers doit être indépendant des autres et peut, par conséquent, être remplacé sans engendrer des modifications dans les autres tiers.

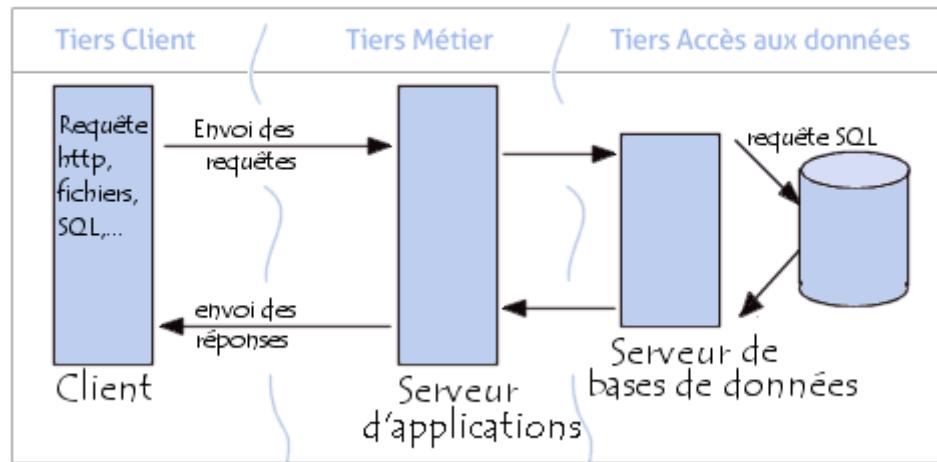


Figure : Architecture du système

Dans le cas de notre projet nous allons présenter une application web, répondant aux besoins des sociétés qui utilisent les applications mailing et pouvant satisfaire à la demande de ses clients.

De ce fait le système à réaliser sera conforme aux modèles d'architectures de 3 tiers ou l'application est divisée en trois niveaux logiques comportant chacun un ensemble d'interfaces bien définies.

Pour bien suivre l'architecture de 3 tiers nous allons détailler dans ce chapitre la conception de la base de données générée.

➤ **Les modèles de cycle de vie :**

Le « cycle de vie d'une application web » désigne toutes les étapes de développement d'une application, de sa conception à sa disparition.

Ces modèles consistent en une succession de phases dont chacune est méthodiquement vérifiée avant de passer à l'étape suivante :

➤ **Conception détaillée**

Cette étape consiste à définir précisément chaque sous-ensemble de notre application Web.

Codage (Implémentation ou programmation)

C'est la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.

Tests unitaires

Ils permettent de vérifier individuellement que chaque sous-ensemble de l'application Web est implémenté conformément aux spécifications.

Intégration

L'objectif est de s'assurer de l'interfaçage des différents éléments (modules) de l'application Web. Elle fait l'objet de tests d'intégration consignés dans un document.

Qualification (ou recette)

C'est-à-dire la vérification de la conformité de l'application Web aux spécifications initiales.

Documentation

Elle vise à produire les informations nécessaires pour l'utilisation de l'application Web et pour des développements ultérieurs.

Mise en production

C'est le déploiement sur l'espace de l'application Web.

Maintenance

Elle comprend toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur l'application Web.

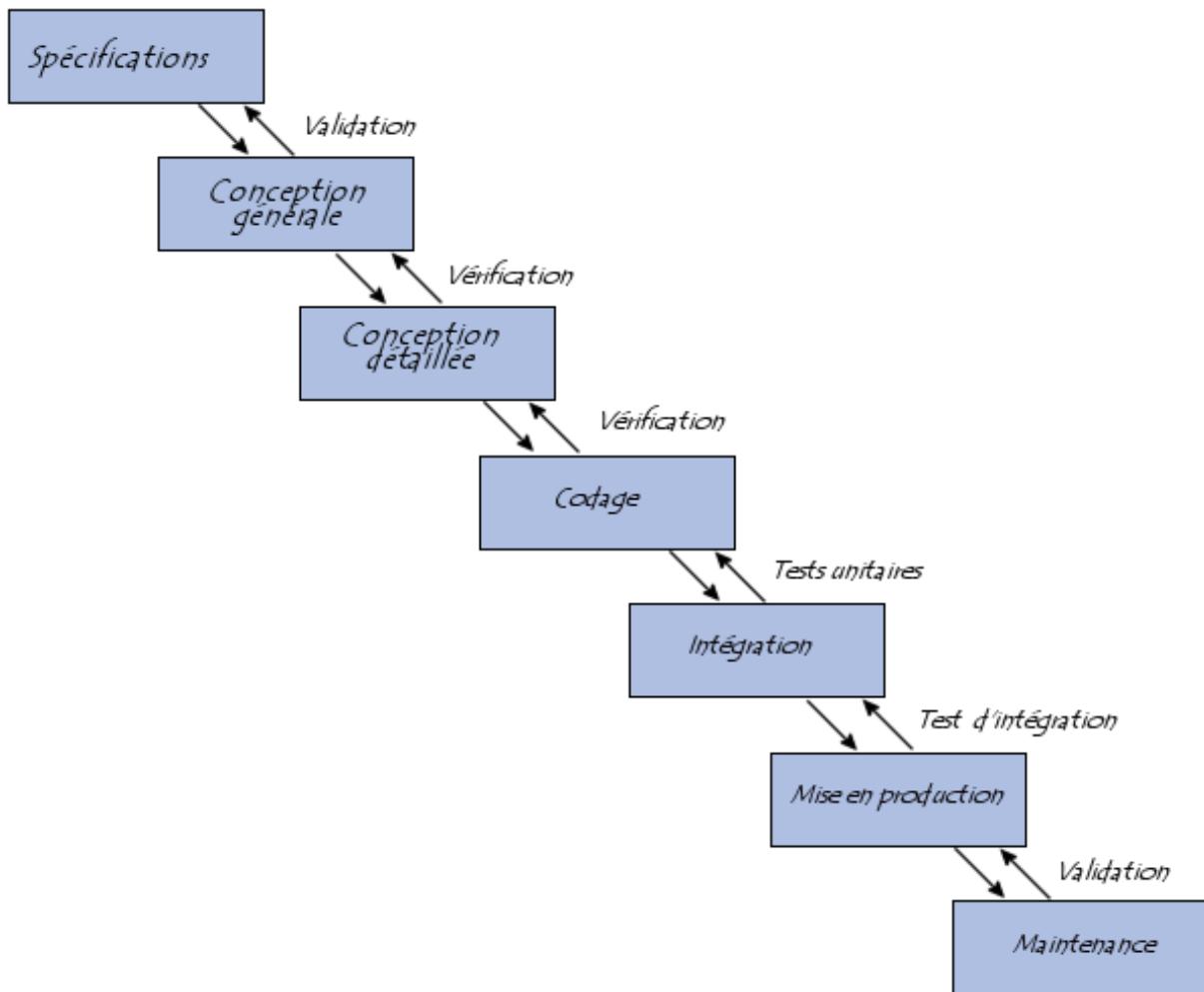


Figure : Modèle du cycle de vie en cascade

III. Choix du langage de conception :

A. Choix d'UML

Par rapport à toutes les méthodes orientées objets qui sont en utilisation, seule UML a la capacité de satisfaire tous les besoins de conceptions requises par les entreprises et les sociétés informatiques

En effet, Il unifie les notations nécessaires aux différentes activités d'un processus de développement et offre en plus de ça les moyens d'établir le suivi des décisions prises depuis les spécifications jusqu'au codage.

Notion d'UML :



UML :(Unified Modeling Language) ou « Langage de Modélisation Unifié » Standardisé par l'OMG (Object Management Group) est une notion basée principalement sur les méthodes des BOOCH (de BOOCH), OMT de Rumbaugh et OOSE de Jacobson.UML a été proposé afin de standardiser les produits de développement (modèle, notation, diagramme) sans standardiser le processus de développement qui dépend des personnes, des applications, des cultures, etc.

UML se propose de créer un langage de modélisation utilisable à la fois par les humains (forme graphique) et les machines (syntaxe précise).

L'utilisation des modèles « UML » sert à :

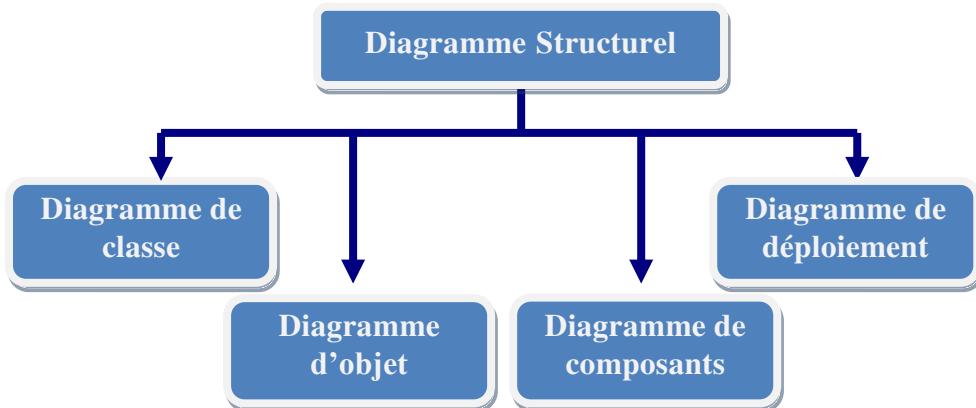
- ✓ Décomposer le processus de développement.
- ✓ Mettre en relation les experts des métiers et les analystes.
- ✓ Coordonner les équipes d'analyse de la réalisation.
- ✓ Séparer l'analyse de la réalisation.
- ✓ Prendre en compte l'évolution de l'analyse et du développement.
- ✓ Migrer facilement vers une architecture objet d'un point de vue statique.

➤ Pourquoi une méthode objet ?

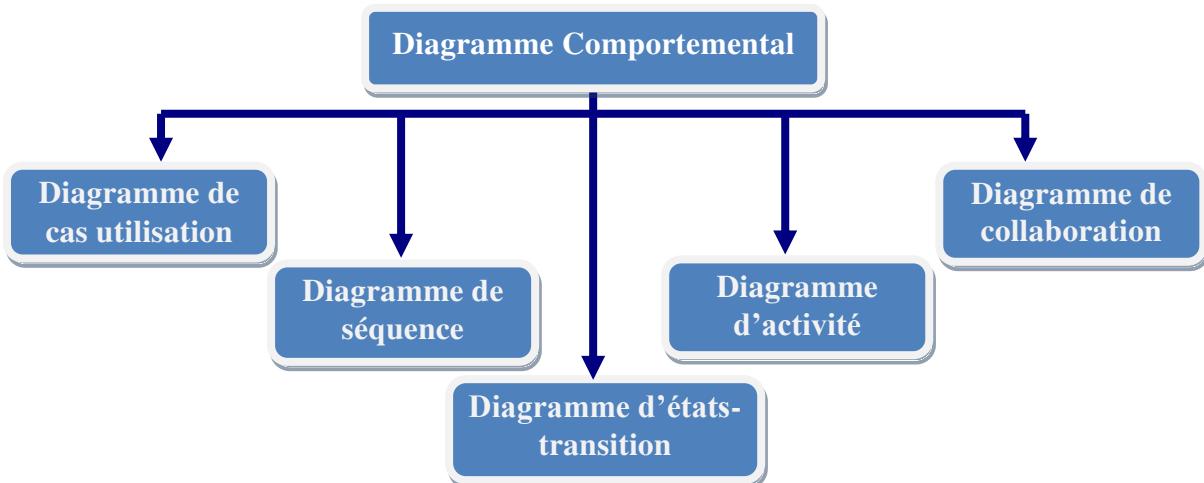
Les langages orientés objet constituent chacun une manière spécifique d'implémenter le paradigme objet. Ainsi, une méthode objet permet de définir le problème à haut niveau sans rentrer dans les spécificités d'un langage. Il représente ainsi un outil permettant de définir un problème de façon graphique, afin par exemple de le présenter à tous les acteurs d'un projet (n'étant pas forcément des experts en un langage de programmation).

UML comporte 9 diagrammes standards représentant autant de "vues" d'un système d'information. Ces diagrammes se répartissent en deux catégories : quatre représentent la structure statique de l'application; cinq représentent son comportement dynamique.

➤ **Vue Statique :**



➤ **Vue Dynamique :**



Ces diagrammes sont d'une utilité variable selon les cas et ils ne sont pas tous nécessairement produits à l'occasion de la modélisation.

Les points forts d'UML :

UML est un langage formel et normalisé

- ✓ Gain de précision
- ✓ Encourage l'utilisation d'outils
- ✓ Gage de stabilité

UML est un support de communication performant

- ✓ Il cadre l'analyse
- ✓ Il facilite la compréhension de représentations abstraites complexes
- ✓ Son caractère polyvalent et sa souplesse en font un langage universel
- ✓ UML a pour objectif de spécifier, construire, visualiser et documenter les systèmes à base de logiciel
- ✓ UML n'est pas une méthode mais une notation qui laisse la liberté de conception.
- ✓ UML est un langage graphique qui permet de modéliser tous les types de systèmes informatiques mais, qui nécessite toute fois une méthodologie de conception (UP, RUP, ..).

Les points faibles d'UML

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

IV. Conception

A. Diagramme de Cas d'utilisation

1. Définition

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système.

Il constitue un de diagrammes le plus structurants dans l'analyse d'un système.

- **Acteur** : représente un rôle joué par une personne qui interagit directement avec le système étudié.
- **Cas d'utilisation (use case)** : représente un ensemble des séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

L'utilisation d'un diagramme de cas d'utilisation s'avère indispensables pour décrire les besoins fonctionnels. Ces diagrammes permettent de décrire l'interaction entre l'acteur et le système.

C'est l'image d'une fonctionnalité de système déclenchée en réponse à la stimulation d'un acteur externe.

2. Concepts de base

❖ Acteurs de système :

Notre application fait intervenir deux acteurs compte tenu du déroulement et de complémentarité des opérations qui sont :

Acteurs	Description
L'administrateur :	<ol style="list-style-type: none"> 1. Authentification () 2. Gestion des utilisateurs : ajouter (), modifier (), supprimer (), bloquer () 3. activer/désactiver l'envoi automatique des messages 4. contrôler et supprimer les messages créés par les clients. 5. gestion des messages envoyés par les clients et les visiteurs : répondre (), supprimer () 6. déconnexion
Client:	<p>Il bénéficie de la plupart des fonctions car il a la possibilité de consulter et de gérer les différents fonctionnalités</p> <ol style="list-style-type: none"> 1. Créer un compte (s'inscrire) 2. Authentification 3. Gestion des messages : composer(), envoyer(), modifier(), supprimer(), 4. Gestion des contacts : Ajouter (), modifier(), supprimer(), importer(), exporter(), gérer contacts de la

	<p>liste noir ()</p> <p>5. Gérer les listes de contact : Ajouter(), modifier() ,supprimer()</p> <p>6. Protéger données d'utilisateur : modifier mot de passe()</p> <p>7. Consulter les statiques d'envoie</p> <p>8. Analyser avec Google Analytics</p> <p>9. Gestion des événements de calendrier : Ajouter (), modifier(), supprimer()</p> <p>10. gérer l'envoi automatique des messages</p> <p>11. contacter l'administrateur</p> <p>12. déconnexion</p>
Visiteur	Contacter l'administrateur pour demande d'information ou autres sujets.

❖ Diagramme de cas d'utilisation initial

Le modèle de cas d'utilisation générale est représenté dans la figure qui va suivre :

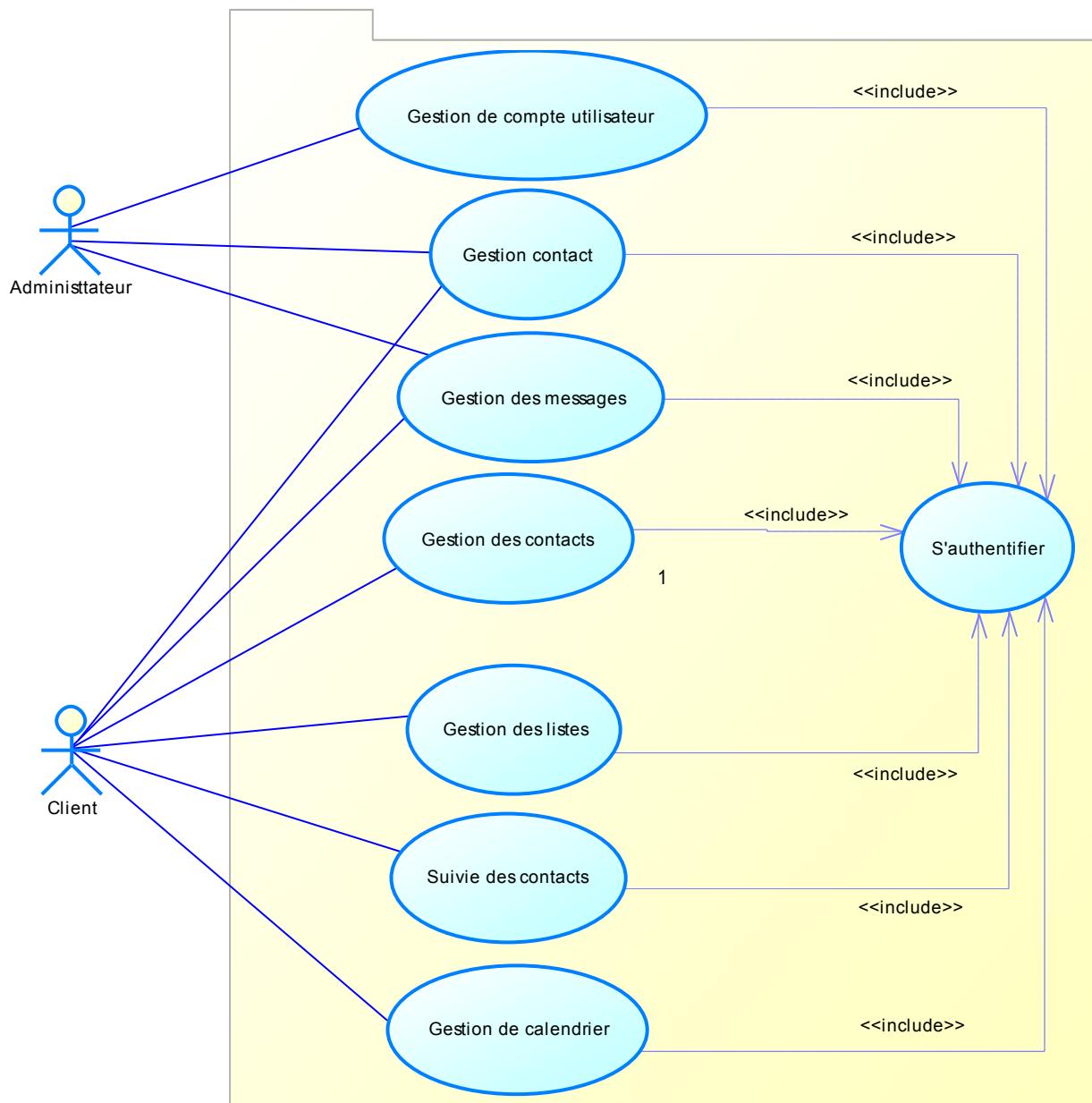


Figure : Diagramme de cas d'utilisation initial

Ce diagramme représente les différentes activités par l'administrateur afin de gérer et contrôler leur application et les activités qui peuvent être effectuées par un client.

❖ Raffinement de cas d'utilisation « Gestion de compte utilisateur » :

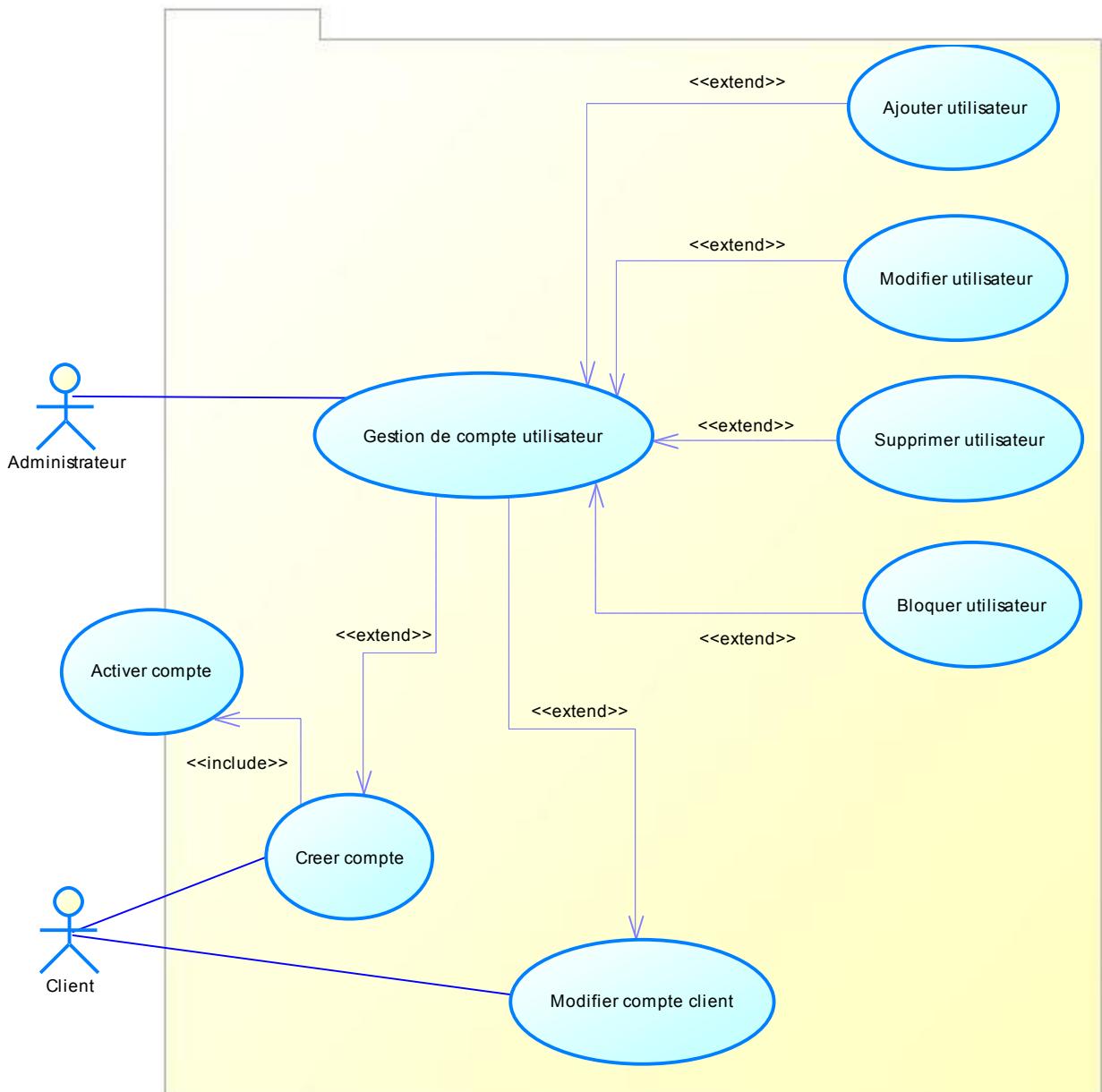


Figure : Diagramme de cas d'utilisation « Gestion de compte utilisateur »

Ce diagramme représente les différents étapes de la gestion de compte utilisateur pour les acteurs administrateur et client :

L'administrateur peut gérer les utilisateurs en effectuant soit l'ajout, soit la modification, soit la suppression ou le blocage.

Le client peut créer un compte qu'une fois valide que lors de l'activation de compte à partir d'un lien qui va reçu par email, il peut aussi protéger leurs données à partir de la modification de mot de passe périodiquement.

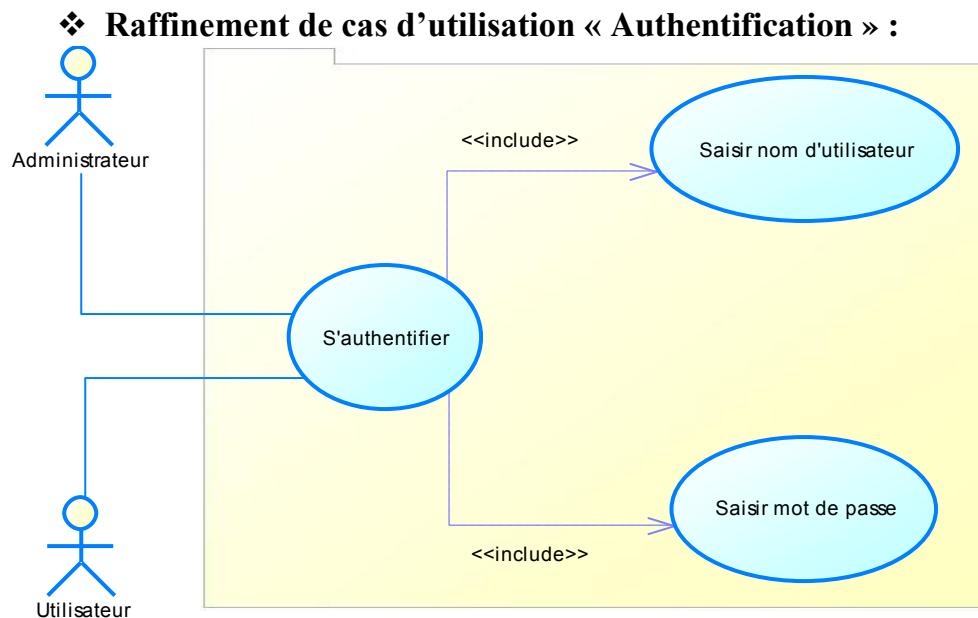


Figure : Diagramme de cas d'utilisation « Authentification »

Ce cas d'utilisation est nécessaire pour chaque utilisateur de notre application.

Ce dernier commence par saisir son nom d'utilisateur et mot de passe, le système vérifie la validité des informations saisies en entrée :

Si les informations saisies sont correctes alors l'utilisateur va rédiger vers la page d'accueil (selon la nature d'utilisateur soit administrateur, soit client).

Si une information est erronée, un message d'erreur s'affichera.

❖ Raffinement de cas d'utilisation « Gestion des messages » :

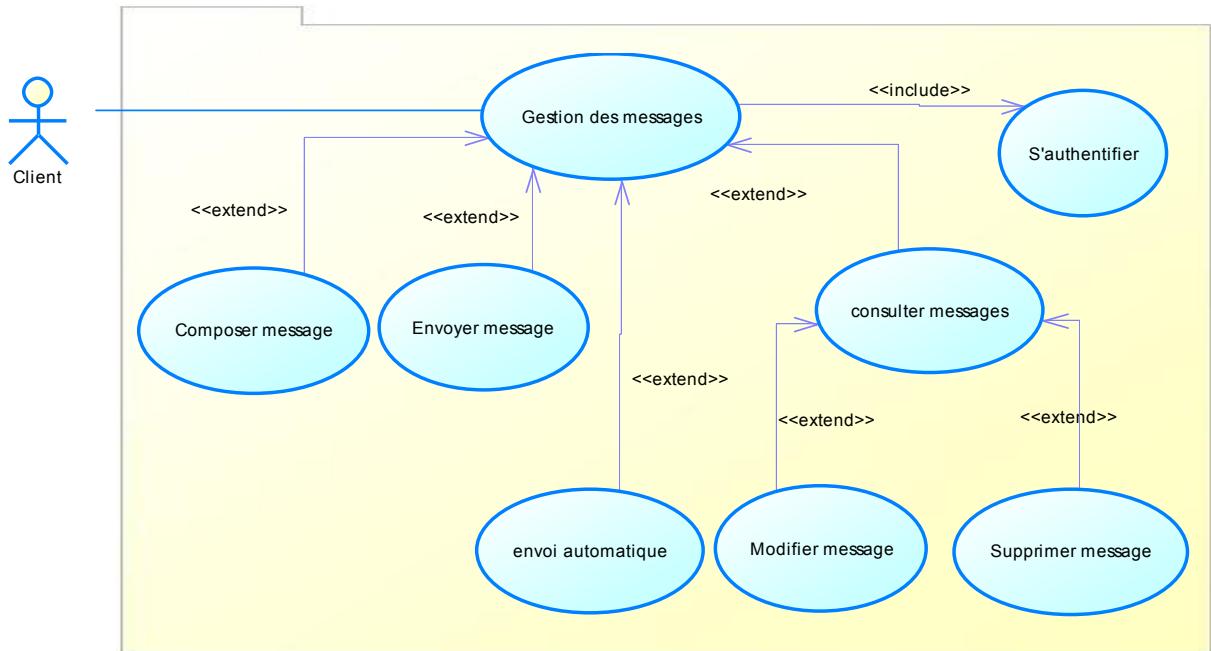


Figure : Diagramme de cas d'utilisation « Gestion des messages »

Le client lors de la gestion des messages, il peut composer, modifier ou supprimer des messages comme il peut envoyer les messages qui existent déjà.

❖ Raffinement de cas d'utilisation « Gestion des contacts » :

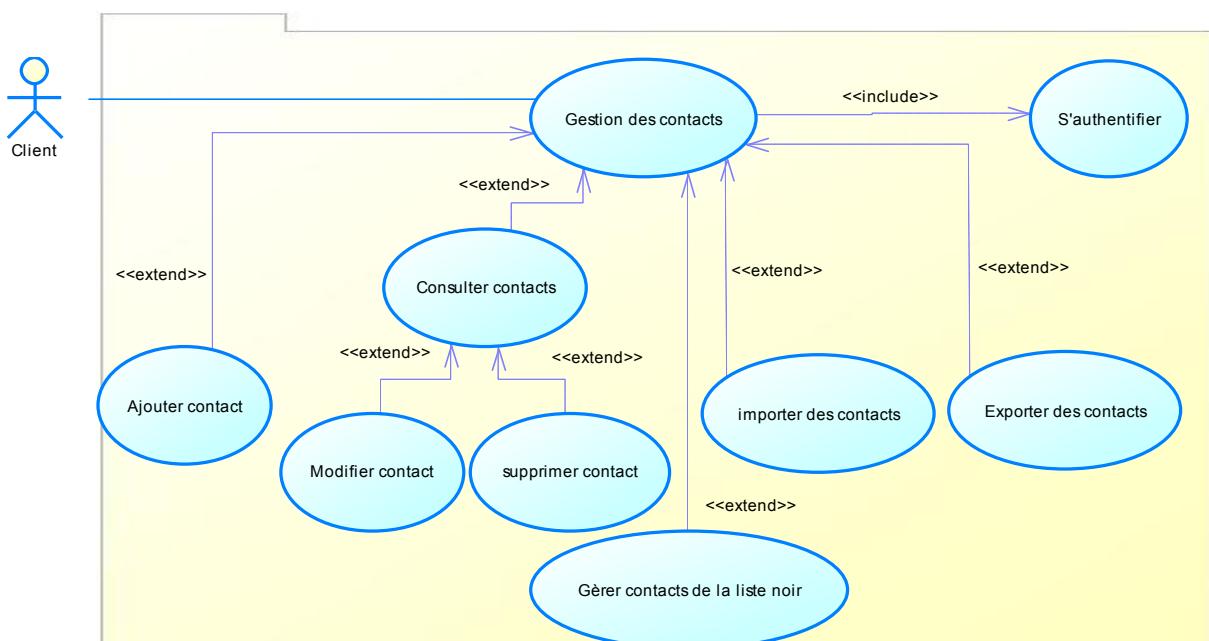


Figure : Diagramme de cas d'utilisation « Gestion des contacts »

Le client peut gérer les contacts, il a le droit d'ajouter, modifier ou supprimer les contacts, il peut aussi les importer ou les exporter ou bien gérer les contacts de la liste noir.

❖ Raffinement de cas d'utilisation « Gestion des listes »:

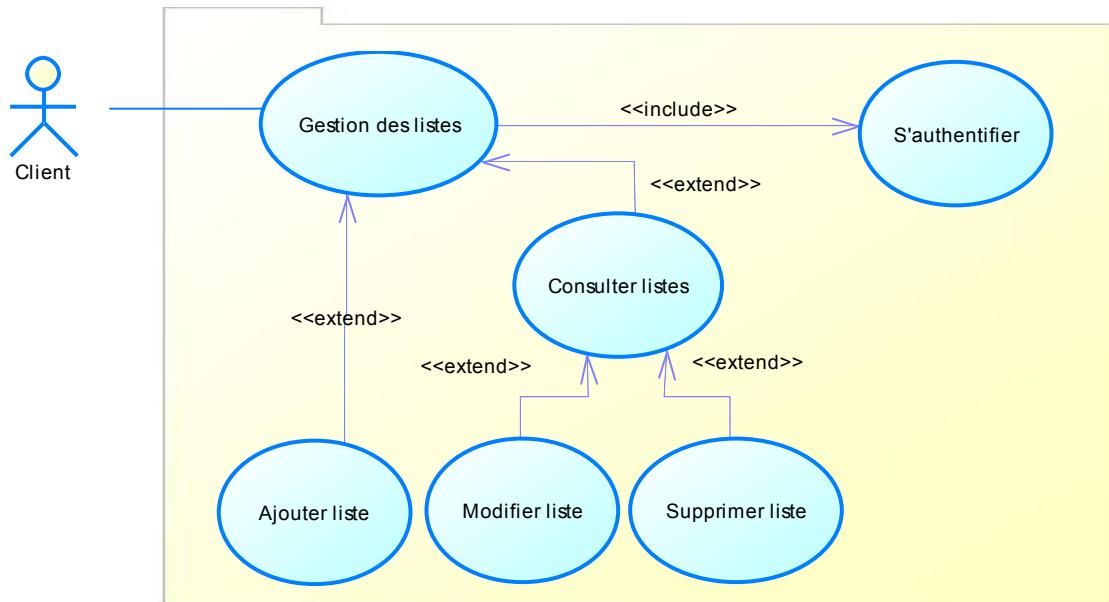


Figure : Diagramme cas d'utilisation « Gestion des listes »

Lors de la gestion des listes, le client peut ajouter, modifier ou supprimer les listes des contacts.

❖ Raffinement de cas d'utilisation « Gestion de calendrier »:

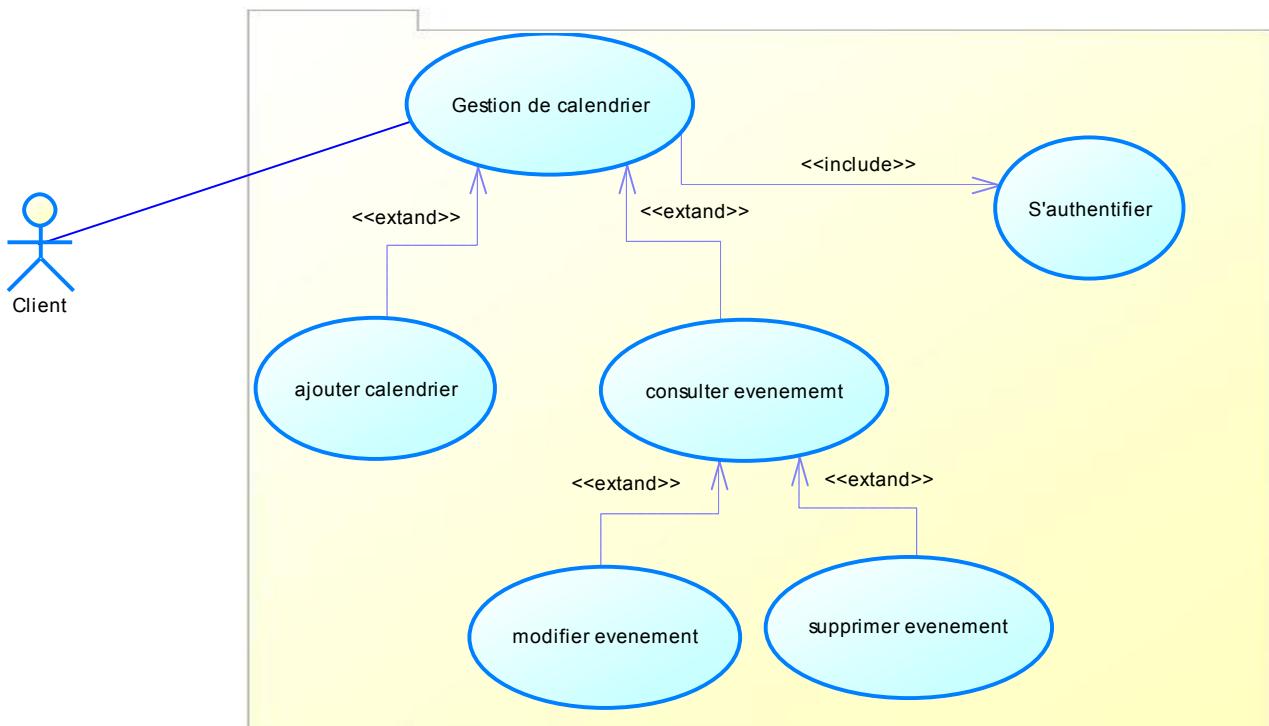


Figure : Diagramme cas d'utilisation « Gestion de calendrier »

Lors de la gestion de calendrier, le client peut ajouter, modifier ou supprimer un événement de calendrier.

❖ **Raffinement de cas d'utilisation « Gestion messages contact »:**

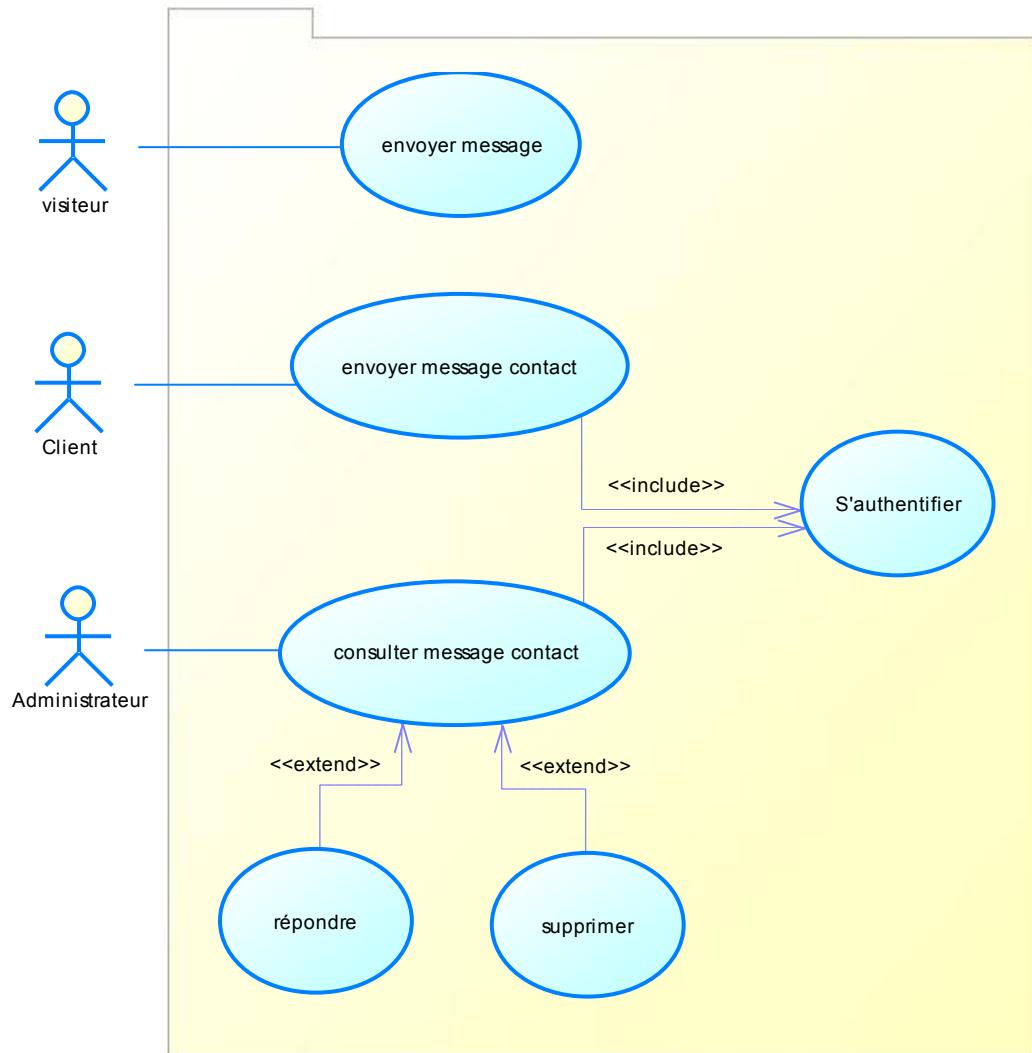


Figure : Diagramme cas d'utilisation « Gestion de messages contact »

Lors de la gestion de messages contact, le visiteur envoie des messages à l'administrateur aussi le client peut envoyer aussi des messages après l'authentification, et l'administrateur consulter ces messages et a le choix de répondre ou de supprimer les messages indésirables.

❖ Cas d'utilisation Final :

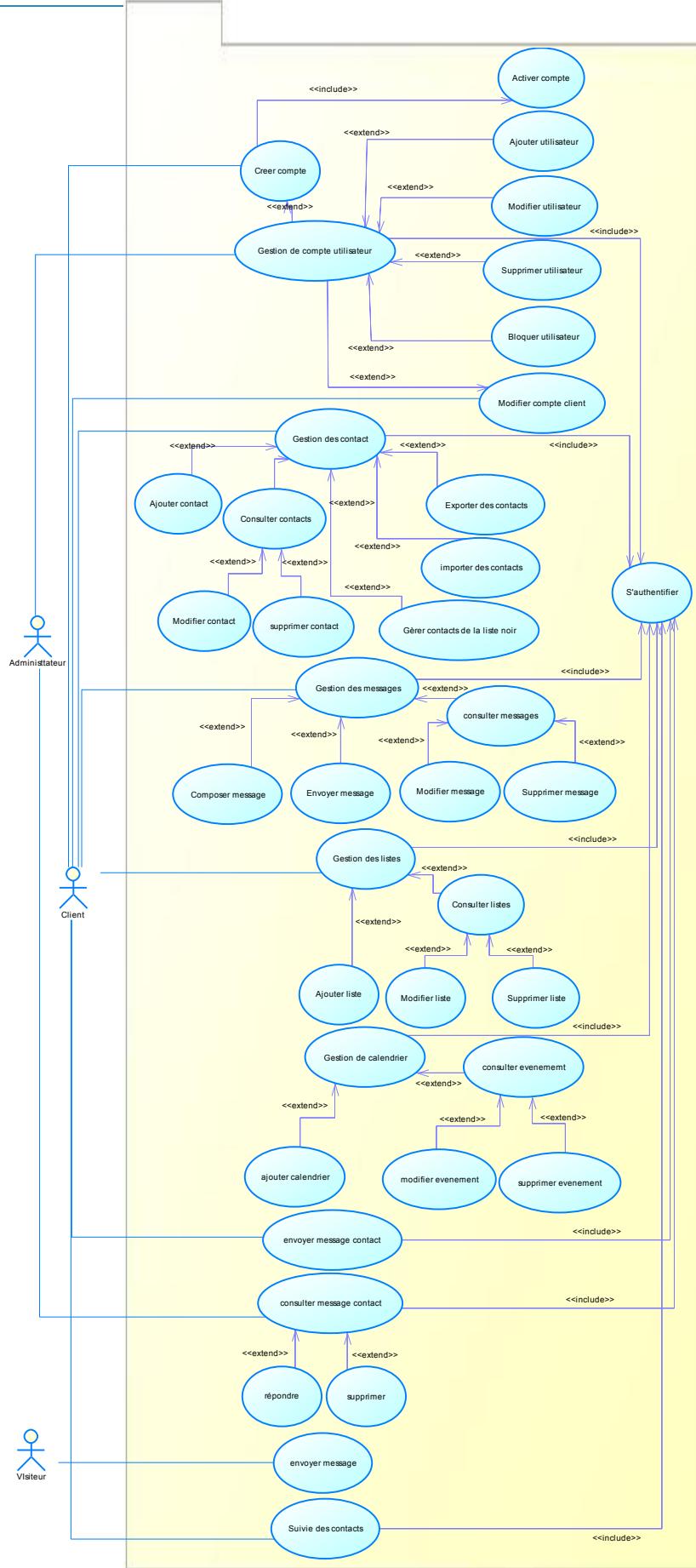


Figure: Cas d'utilisation « Final »

B. Diagramme de séquence :

Définition de diagramme de séquence :

Les diagrammes de séquence montrent la séquence des interactions entre objets selon un point de vue temporel (chronologique). Ce diagramme permet de représenter les scénarios d'un cas d'utilisation, il permet de mieux visualiser la séquence des messages par une lecture de bas en haut .Un scénario est une instance d'un cas d'utilisation [7].

❖ Diagramme de séquence d' « Inscription » :

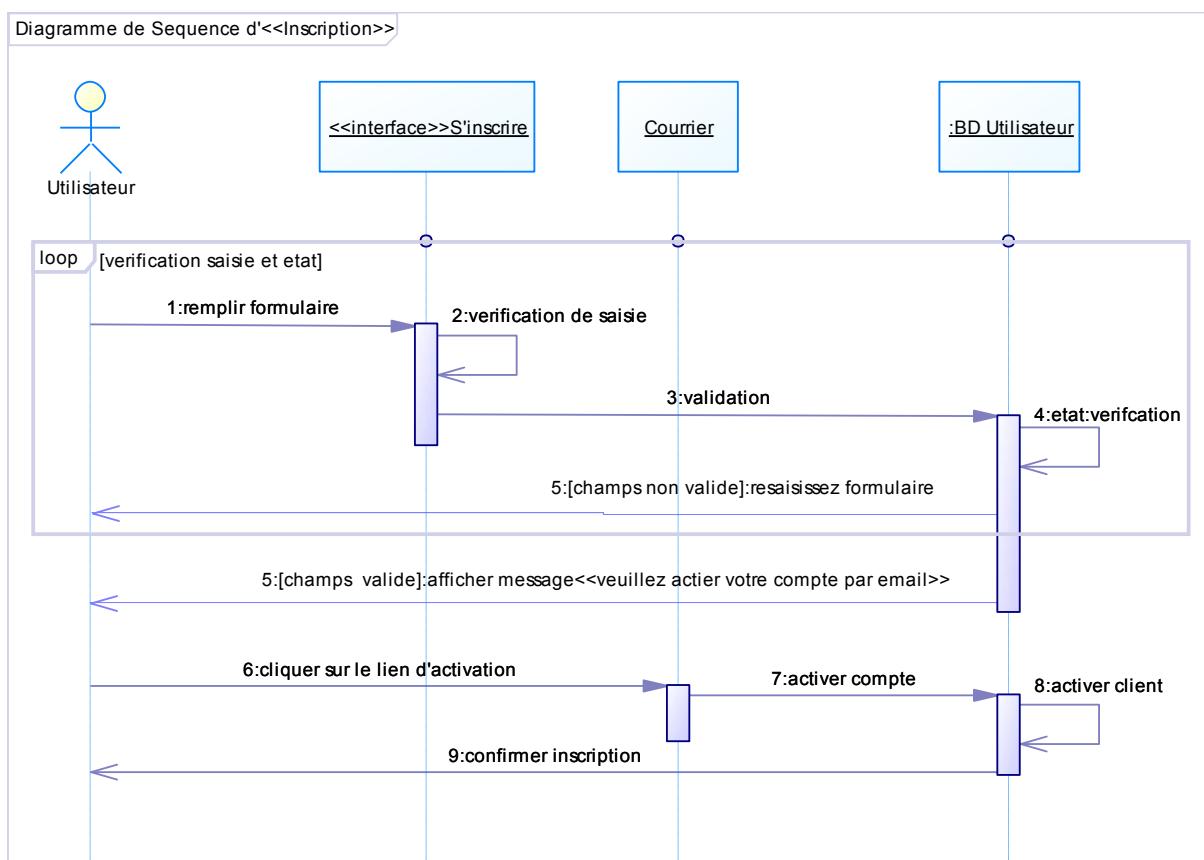


Figure : Diagramme de séquence d' « Inscription »

Pré condition :

Opération demandé par l'utilisateur.

Scénario normale :

1. L'utilisateur doit remplir le formulaire de l'interface inscription.
2. le système contrôle les champs saisis.
3. l'utilisateur valide la saisie.
4. le système vérifie l'existence de l'utilisateur: Si les champs non valide, il faut resaisisse le formulaire
5. si les champs valides le système affiche un message «veuillez activer votre compte par email».

6. l'utilisateur clique sur le lien d'activation
7. le système activé le compte utilisateur
8. le compte client est activé
9. confirmation d'inscription

Post condition :

Utilisateur inscrit.

❖ **Diagramme de séquence d' « Authentification » :**

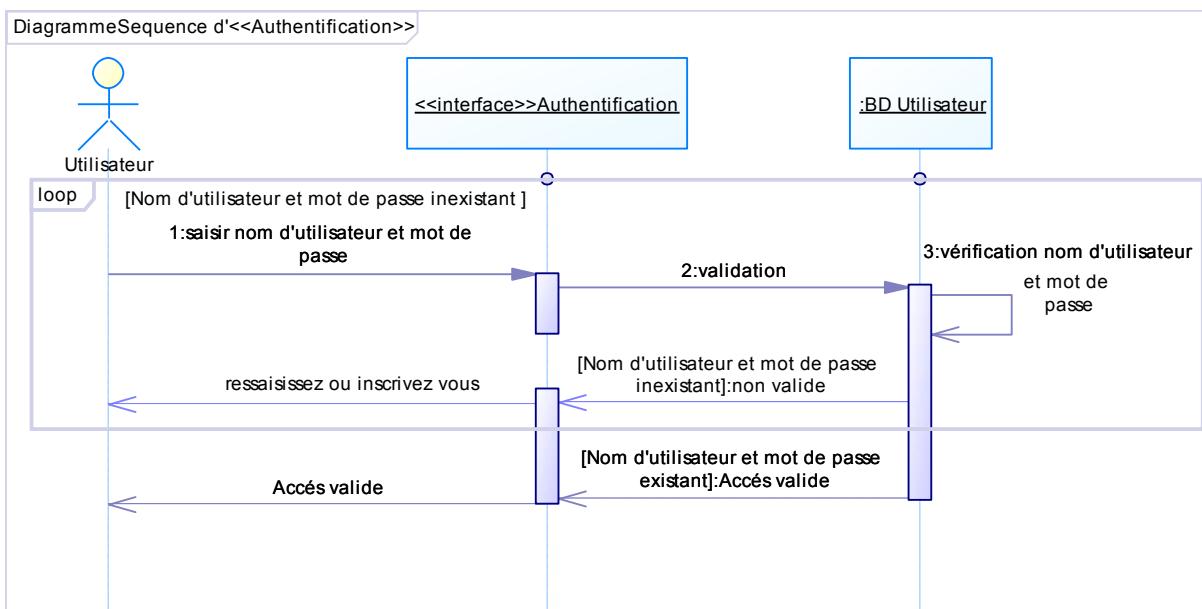


Figure : Diagramme de séquence « Authentification »

Pré condition :

L'opération d'authentification doit être affichée.

Scénario normale :

- 1 : L'utilisateur doit saisir son nom d'utilisateur et mot de passe et demande l'accès à l'application.
- 2 : Envoyer de nom d'utilisateur et mot de passe à la base des données.
- 3 : Un test doit être réalisé, celui d'existence et compatibilité du nom d'utilisateur et mot de passe.
- 4 : Si les données correctes et le nom d'utilisateur et mot de passe existe, alors l'utilisateur peut accéder à l'application.
- 5 : Si les données ne sont pas correctes et le nom d'utilisateur et mot de passe inexistant, l'utilisateur doit soit ressayer le saisie, soit inscrire.

Post condition :

L'utilisateur identifié.

❖ **Diagramme de séquence de client « Composer message »:**

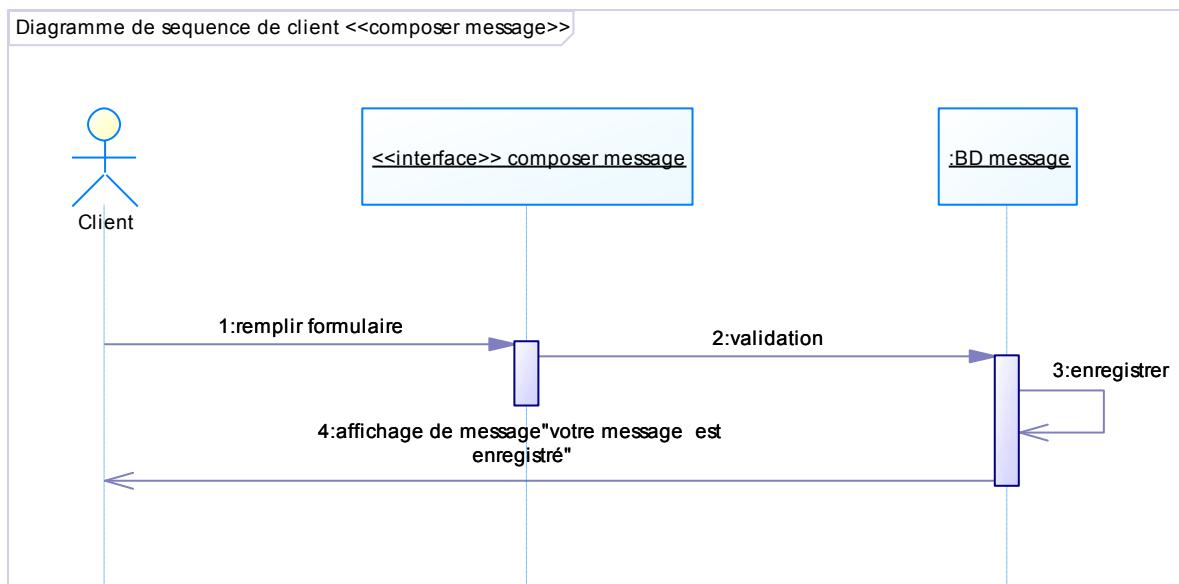


Figure : Diagramme de séquence « Composer message »

Pré condition :

L'opération de composer un message doit être demandé.

Le client doit être authentifié.

Scénario normal :

- 1 : Le client doit remplir la formulaire de l'interface composer message.
- 2 : Le client valide la saisie : Envoi des données à la base des données.
- 3 : Enregistrement des données dans la table message de notre base de données.
- 4 : Le système affiche un message «votre message est enregistré ».

Post condition :

L'ajout de message est effectué.

❖ **Diagramme de séquence de client « Modifier message » :**

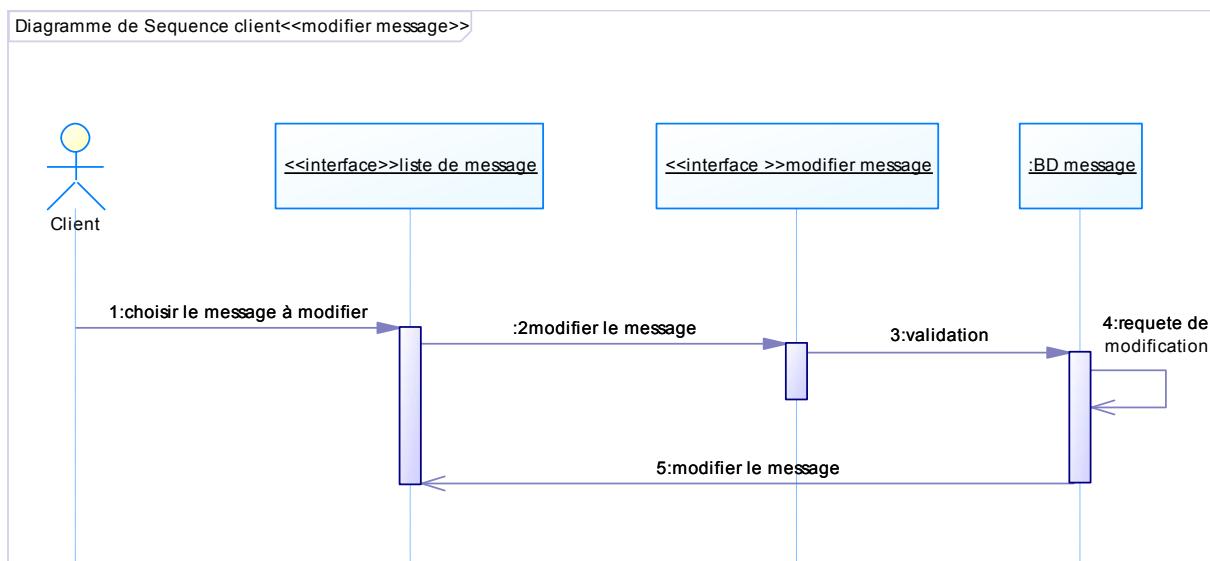


Figure : Diagramme de séquence « Modifier message »

Pré condition :

Le client doit être authentifié.

Scénario normal :

- 1 : Le client choisit le message à modifier ses informations.
- 2 : Le système dirige l'interface de liste des messages vers modifier message.
- 3 : Le client saisie les informations à modifier du message et valide le saisie.
- 4 : Le système exécute la requête de modification et enregistrer les informations modifiés.
- 5 : L'opération de modification de message est effectuée et sera affiché à la liste des messages.

Post condition :

Les modifications sont enregistrées.

❖ **Diagramme de séquence de client « supprimer message »:**

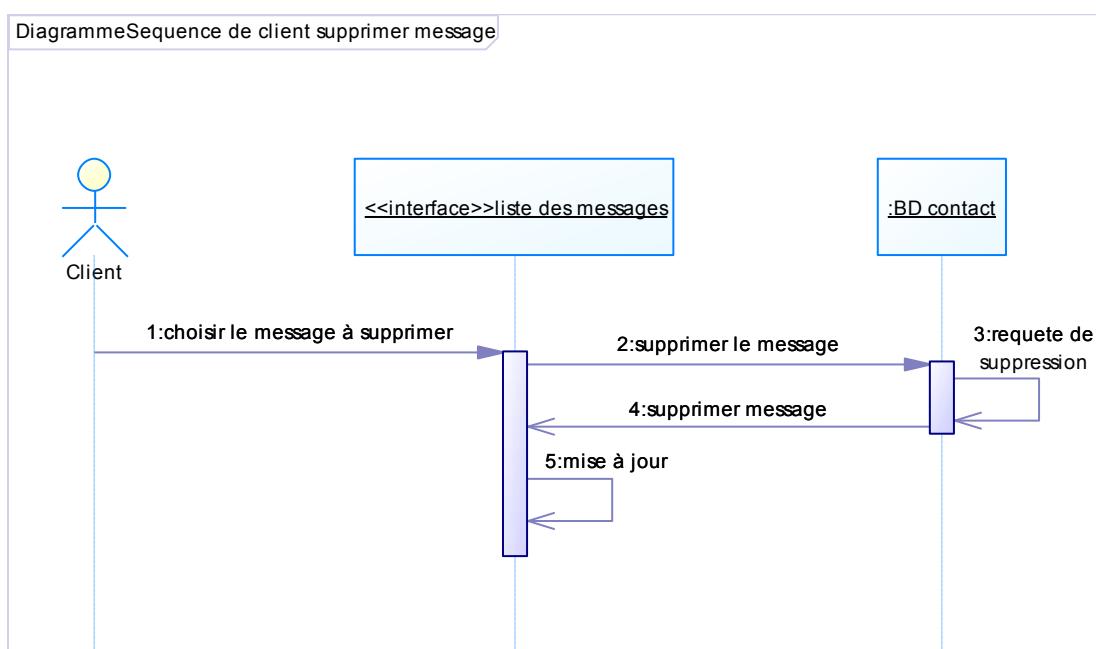


Figure : Diagramme de séquence « Supprimer message »

Pré condition :

Le client doit être authentifié.

Scénario normal :

1. le client doit choisir le message à supprimer.
2. le système affiche un message pour la validation de suppression.
3. Le système exécute la requête de suppression et supprime le message de la base de données.
4. le message est supprimé avec succès.
5. le mise à jour est effectuée à l'interface liste de message.

Post condition :

Le message est supprimé.

❖ **Diagramme de séquence de client « Envoyer message » :**

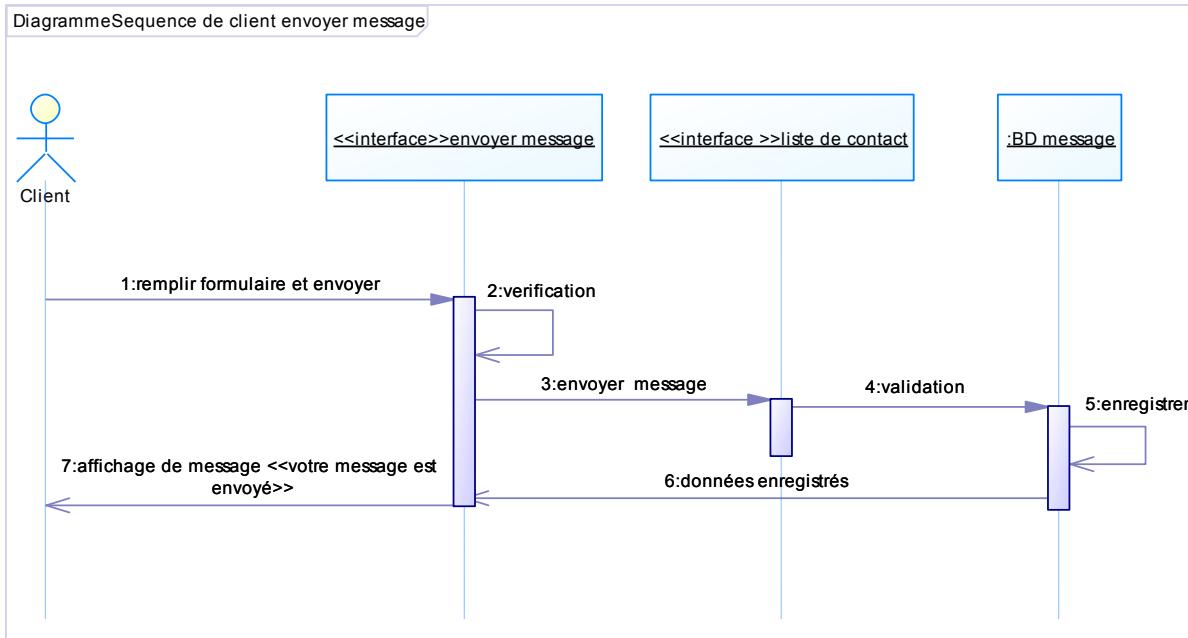


Figure : Diagramme de séquence<<Envoyer message >>

Pré condition :

Le client doit être identifié.

Scénario normal :

- 1 : le client doit remplir et envoyer la formulaire.
- 2 : le système vérifie les champs saisis.
- 3 : le message est bien envoyé à une liste de contacts.
- 4 : une validation de données (état : envoyé et date d'envoi) doit effectué à la table message de la base de données.
- 5 : enregistrement de l'état et de date d'envoi dans la table message.
- 6 : données enregistrés avec succès.
- 7 : affichage de message <<votre message est envoyé>>

Post condition :

Le message est envoyé.

❖ **Diagramme de séquence de client « Ajouter contact » :**

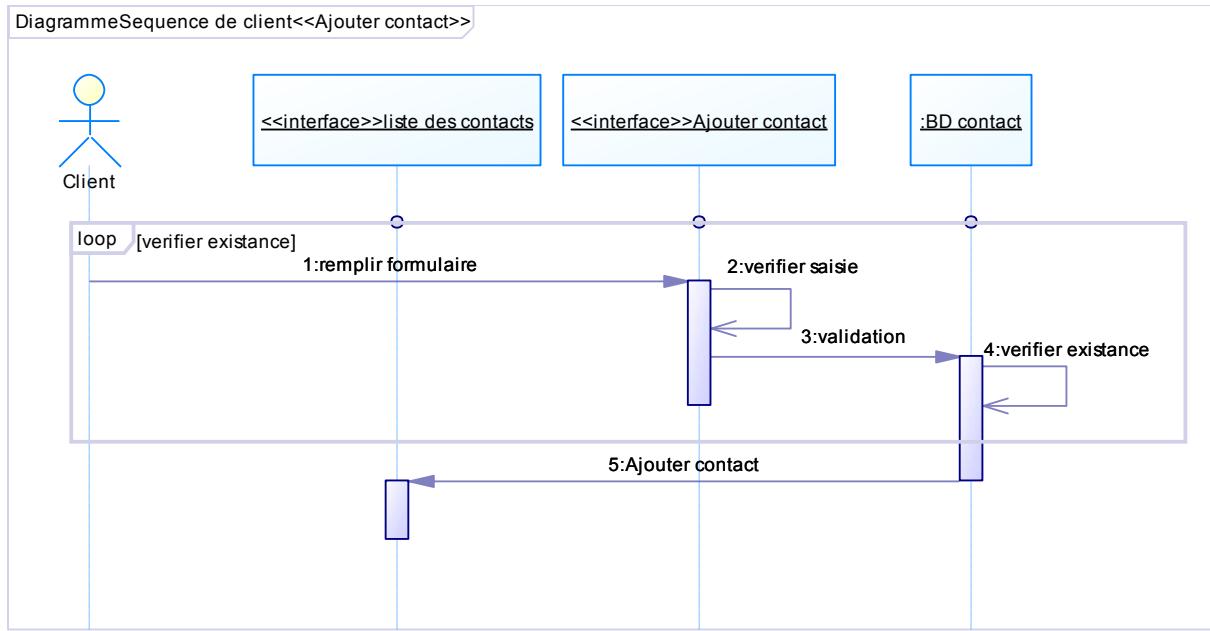


Figure : Diagramme de séquence « Ajouter contact »

Pré condition :

L'opération d'ajout doit être demandée.
Le client doit être authentifié.

Scénario normal :

1. le client doit saisir les informations de client à ajouter.
2. le système contrôle les champs saisis.
3. le client valide la saisie.
4. le système vérifie l'existence de contact : Si les informations existent déjà, le système affiche un message «ce contact existe déjà».
5. Si le contact n'existe pas le système effectue l'ajout de contact.

Post condition :

L'ajout de contact est effectué.

❖ Diagramme de séquence de client « Modifier contact » :

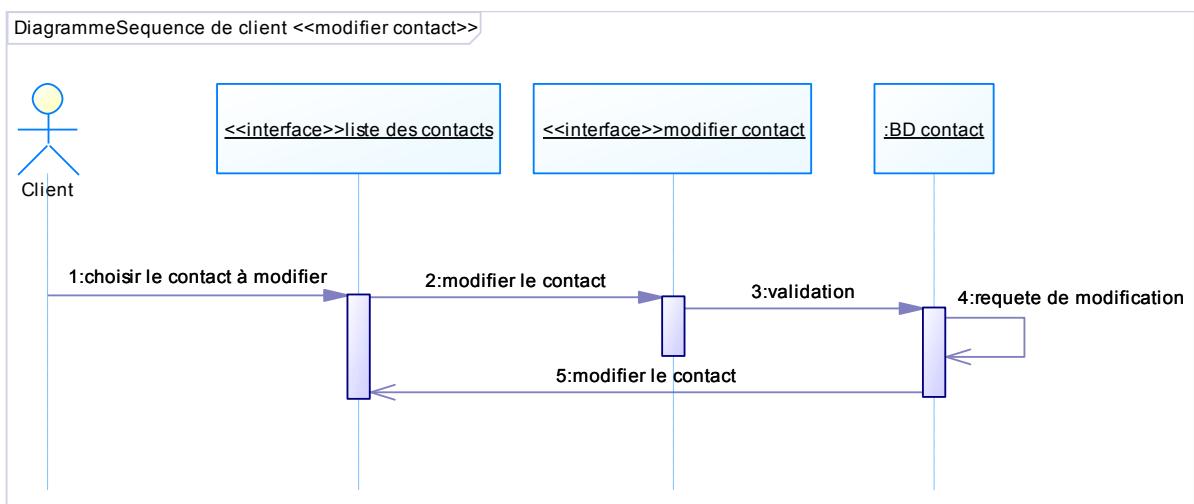


Figure : Diagramme de séquence « Modifier contact »

Pré condition :

Le client doit être authentifié.

Scénario normal :

- 1 : Le client choisit le contact à modifier ses informations.
- 2 : Le système dirige l'interface de liste des contacts vers modifier contact.
- 3 : Le client saisie les informations à modifier du contact et valide le saisie.
- 4 : Le système exécute la requête de modification et enregistrer les informations modifiés.
- 5 : L'opération de modification de contact est effectuée et sera affiché à la liste des contacts.

Post condition :

Le contact est modifié.

❖ Diagramme de séquence de client « Supprimer contact »:

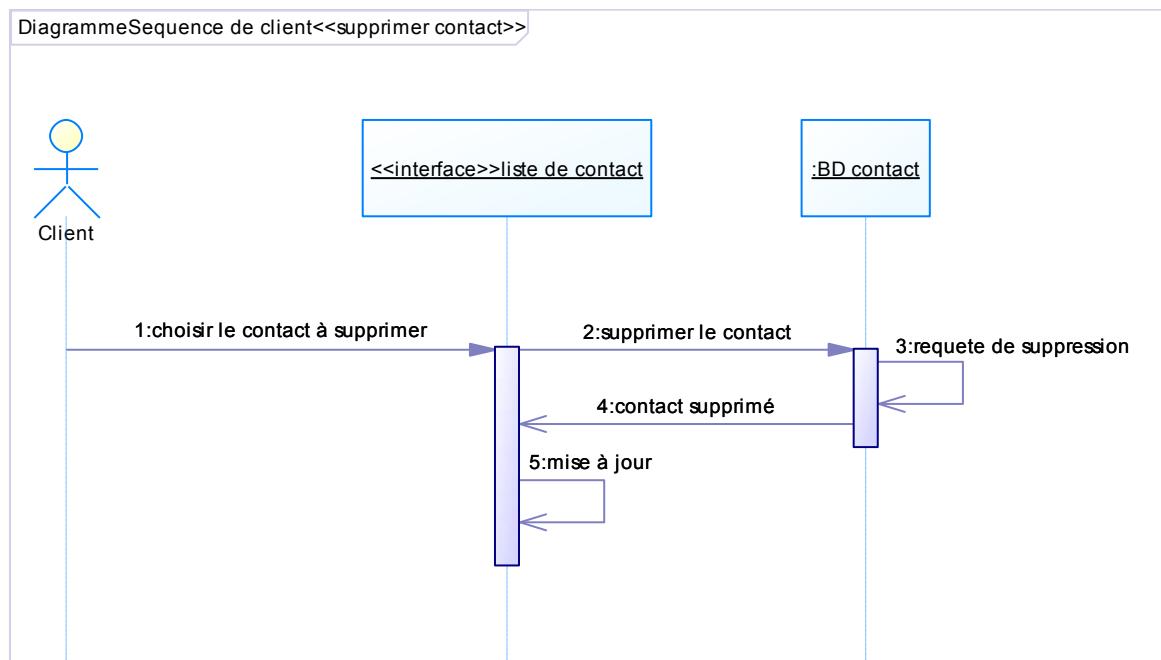


Figure : Diagramme de séquence « Supprimer contact »

Pré condition :

Le client doit être authentifié.

Scénario normal :

1. le client doit choisir le contact à supprimer.
2. le système affiche un message pour la validation de suppression.
3. le système exécute la requête de suppression et supprime le contact de la base de données.
4. le contact est supprimé avec succès.
5. le mise à jour est effectuée à l'interface liste des contacts.

Post condition :

Le contact est supprimé.

❖ Diagramme de séquence de client « gérer contact de la liste noir » :

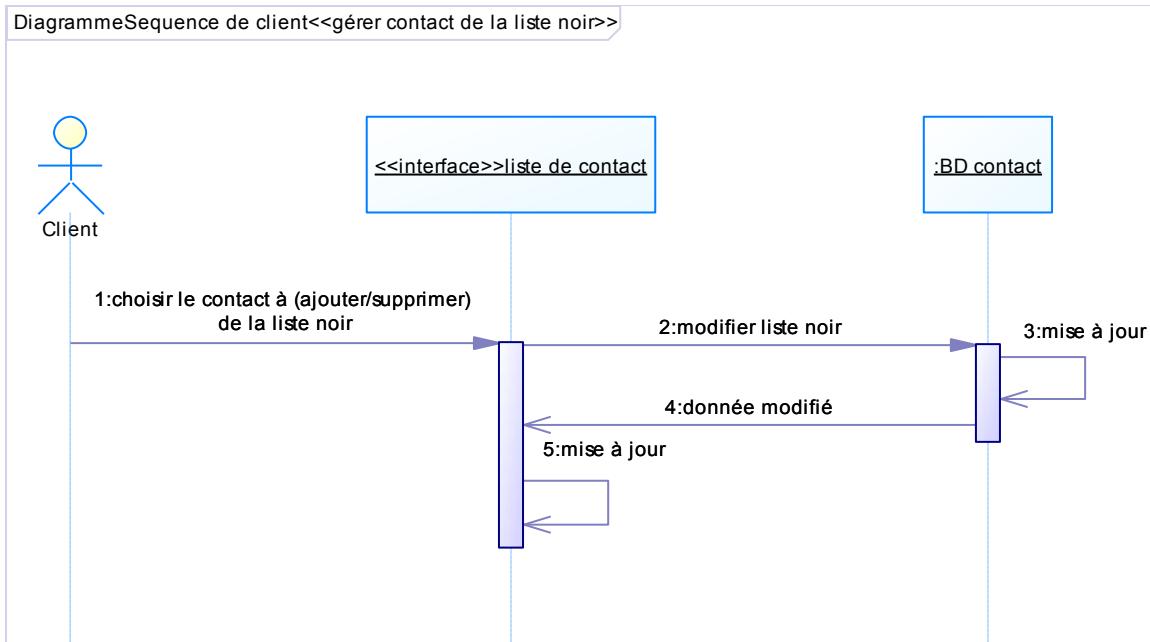


Figure : Diagramme de séquence « gérer contact de la liste noir contact »

Pré condition :

Le client doit être authentifié

Scénario normal :

1. le client doit choisir le contact à ajouter ou supprimer de la liste noir à partir de la liste des contacts.
2. le client valide la modification.
3. une mise à jour sera effectuée au niveau de la table contact de la base des données.
4. les données sont modifiées avec succès.
5. mise à jour des données au niveau de l'interface liste des contacts pour effectué le modification.

Post condition :

Les modifications sont enregistrées.

❖ **Diagramme de séquence de client « Importer contacts » :**

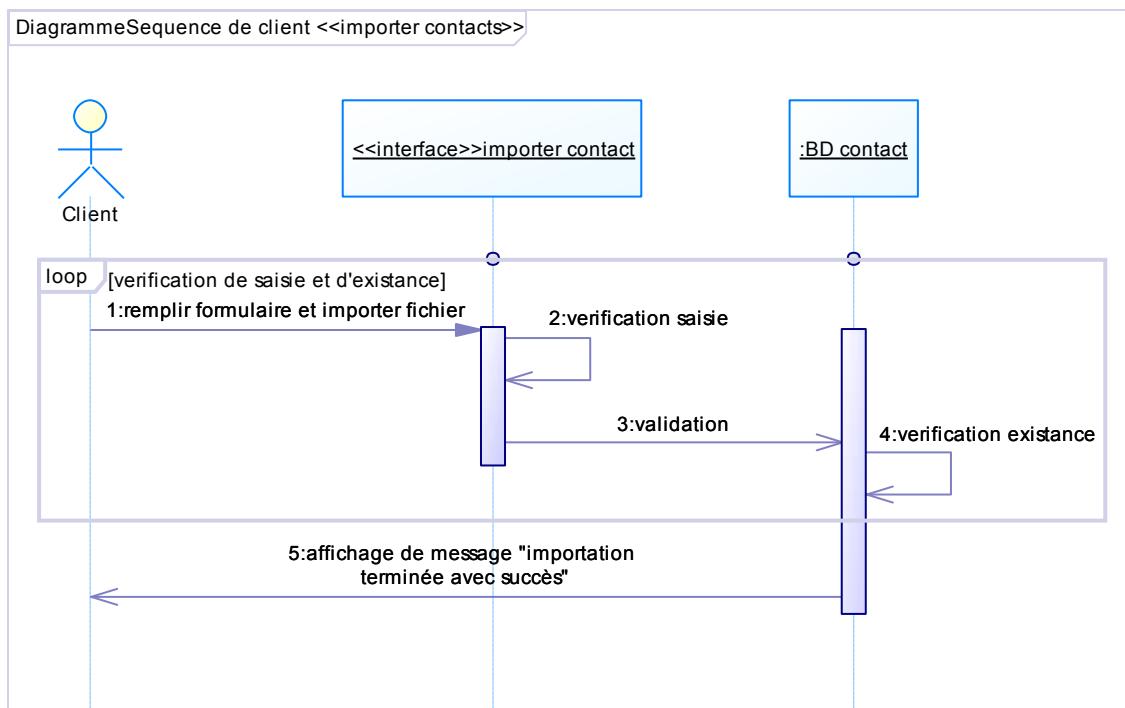


Figure : Diagramme de séquence « Importer contacts »

Pré condition :

L’opération d’importation doit être demandée.

Le client doit être authentifié.

Scénario normal :

1. le client doit remplir le formulaire et doit importer le fichier.
2. le système contrôle les champs saisis.
3. le client valide la saisie.
4. le système vérifie l’existence des contacts dans la base de données : Si les informations existent déjà le système affiche un message «ce contact existe déjà >>
5. si les informations n’existent pas le système effectué l’importation des contacts et affiche un message « importation terminé avec succès »

Post condition :

L’importation des contacts est effectuée.

❖ **Diagramme de séquence de client « Exporter contacts » :**

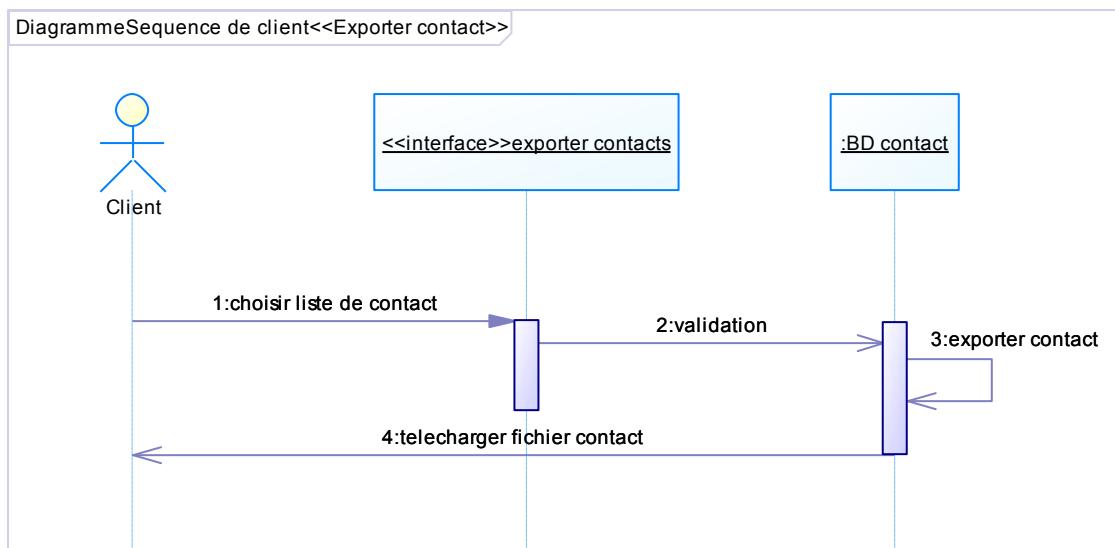


Figure : Diagramme de séquence « Exporter contacts »

Pré condition :

L'opération d'exportation doit être demandée.

Le client doit être authentifié.

Scénario normal :

1. le client doit choisir la liste des contacts à exporter.
2. le client valide l'export.
3. le système exporte les contacts.
4. un fichier qui contient les contacts de la liste choisie par le client sera téléchargé.

Post condition :

L'exportation des contacts est effectuée.

❖ **Diagramme de séquence de client « Ajouter liste des contacts » :**

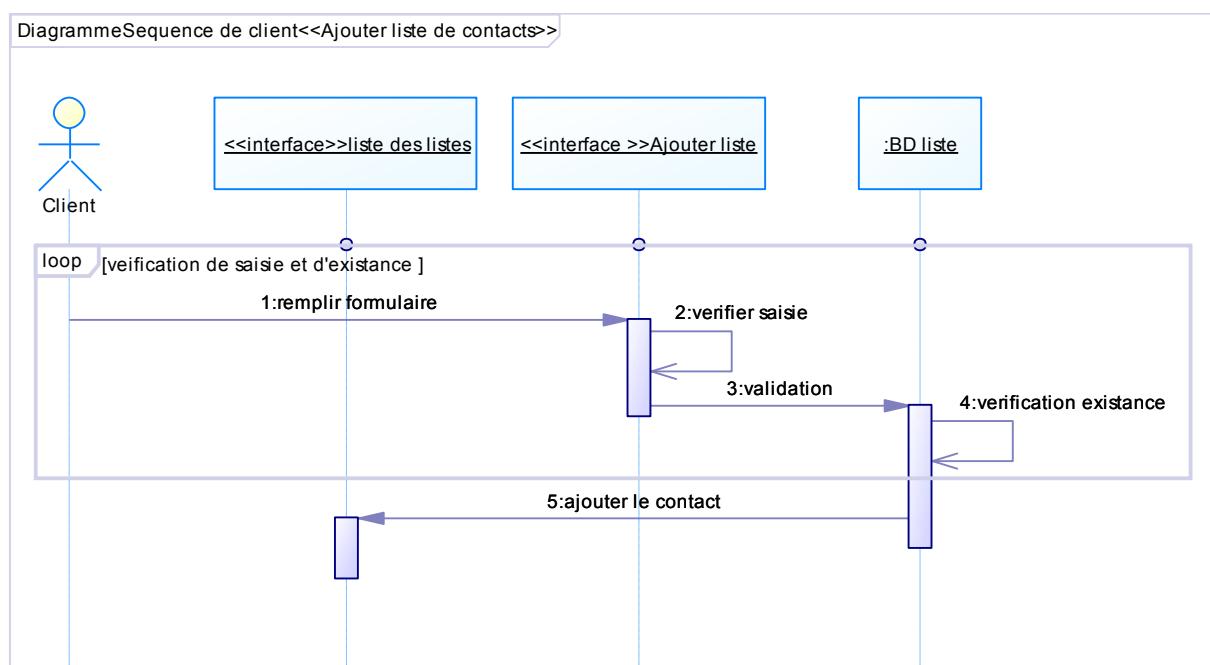


Figure : Diagramme de séquence « Ajouter liste de contacts »

Pré condition :

L'opération d'ajout doit être demandée.

Le client doit être authentifié.

Scénario normal :

1. le client doit être saisi les informations de client à ajouter.
2. le système contrôle les champs saisis.
3. le client valide la saisie.
4. le système vérifie l'existence de liste : Si les informations existent déjà, le système affiche un message «ce liste existe déjà ».
5. Si la liste n'existe pas le système effectue l'ajout de liste.

Post condition :

L'ajout de liste est effectué.

❖ **Diagramme de séquence de client « modifier liste des contacts » :**

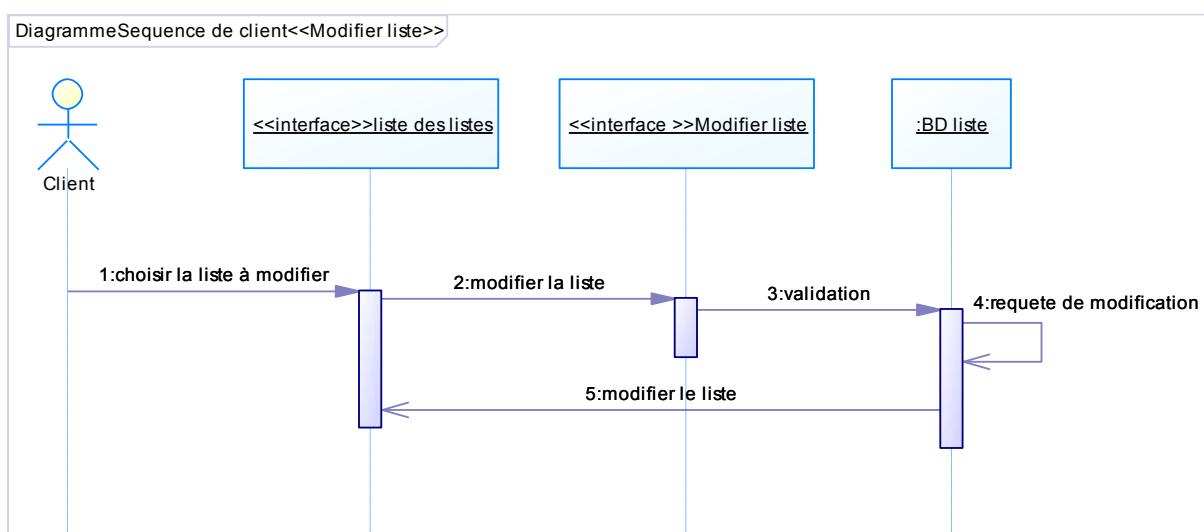


Figure : Diagramme de séquence « Modifier liste des contacts »

Pré condition :

Le client doit être authentifié.

Scénario normal :

- Le client choisit le liste à modifier.
- Le système dirige l'interface de liste des contacts vers modifier liste.
- Le client saisie les informations à modifier liste et valide le saisie.
- Le système exécute la requête de modification et enregistrer les informations modifiés.
- L'opération de modification de liste est effectuée et sera affiché à la liste des listes.

Post condition :

La liste est modifiée.

❖ **Diagramme de séquence de client « supprimer liste des contacts » :**

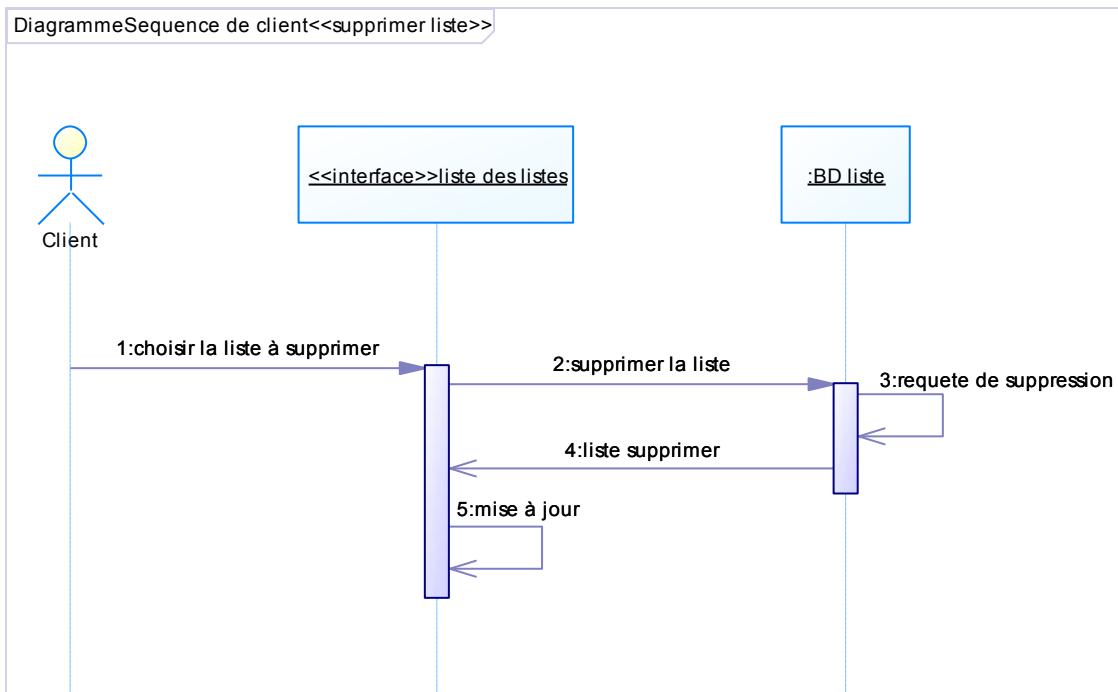


Figure : Diagramme de séquence « Supprimer liste des contacts »

Pré condition :

Le client doit être authentifié.

Scénario normal :

1. le client doit choisir la liste à supprimer.
2. le système affiche un message pour la validation de suppression.
3. le système exécute la requête de suppression et supprime la liste de la base de données.
4. la liste est supprimée avec succès.
5. la mise à jour est effectuée à l'interface liste des listes.

Post condition :

La liste est supprimée.

❖ **Diagramme de séquence d'administrateur « Ajouter client » :**

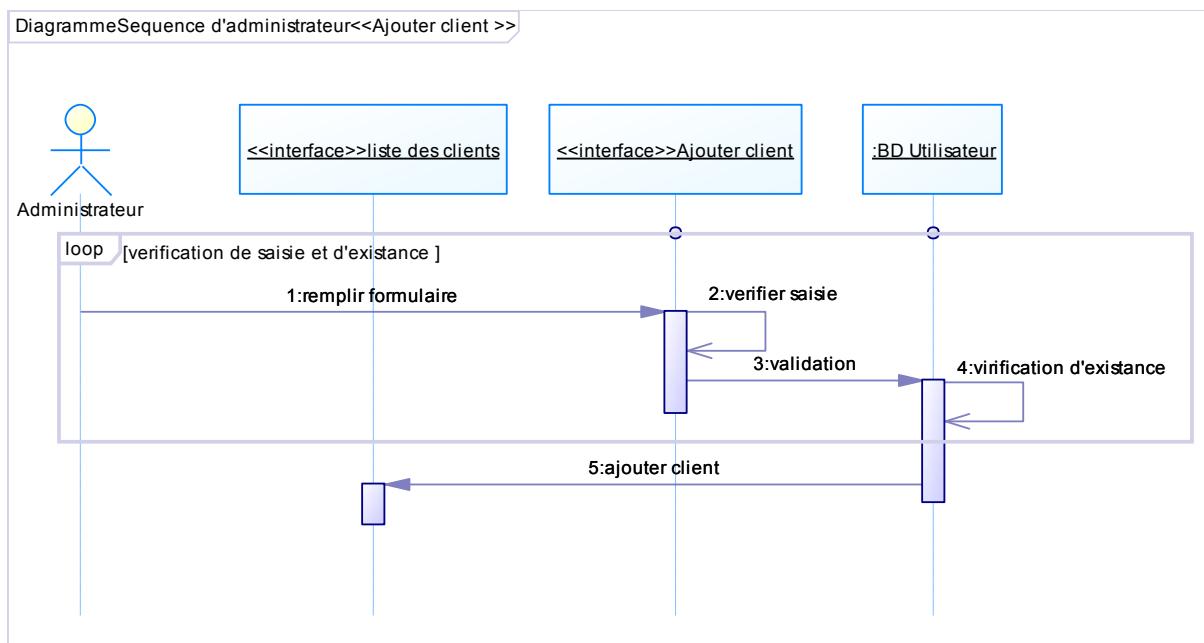


Figure : Diagramme de séquence d'administrateur « Ajouter client »

Pré condition :

L'opération d'ajout doit être demandée.

L'administrateur doit être authentifié.

Scénario normal :

1. L'administrateur doit être saisi les informations de client à ajouter.
2. le système contrôle les champs saisis.
3. l'administrateur valide la saisie.
4. le système vérifie l'existence de client : Si les informations existent déjà, le système affiche un message «ce client existe déjà».
5. Si le contact n'existe pas le système effectué l'ajout de contact.

Post condition :

L'ajout de client est effectué.

❖ **Diagramme de séquence d'administrateur « Modifier client » :**

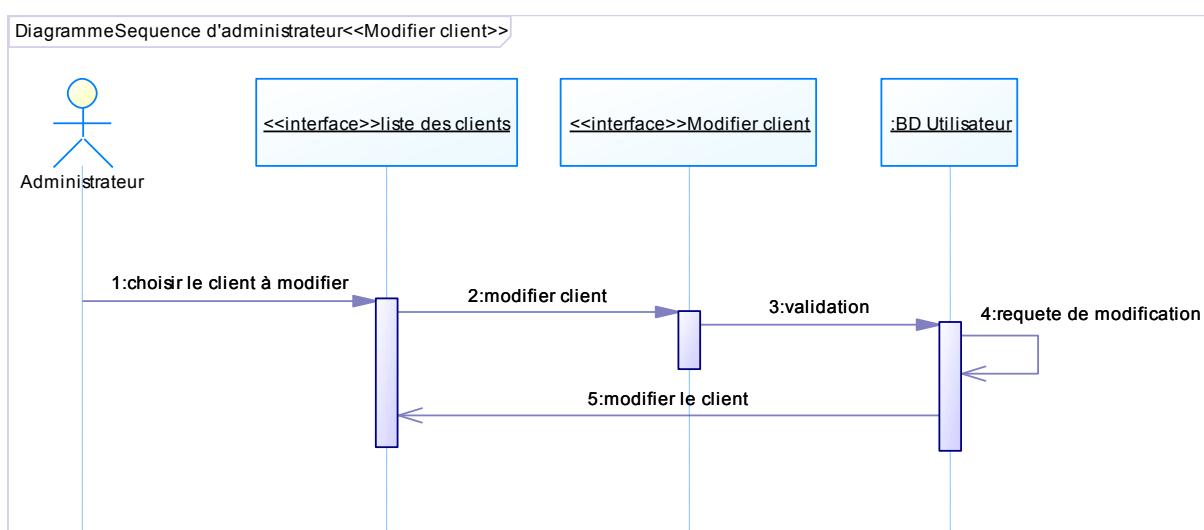


Figure : Diagramme de séquence d'administrateur « Modifier client »

Pré condition :

L'administrateur doit être authentifié.

Scénario normal :

- L'administrateur choisit le client à modifier ses informations.
- Le système dirige l'interface de liste des clients vers modifier client.
- L'administrateur saisie les informations à modifier du client et valide le saisie.
- Le système exécute la requête de modification et enregistrer les informations modifiés.
- L'opération de modification de client est effectuée et sera affiché à la liste des clients.

Post condition :

Le client est modifié.

❖ Diagramme de séquence d'administrateur « Supprimer client » :

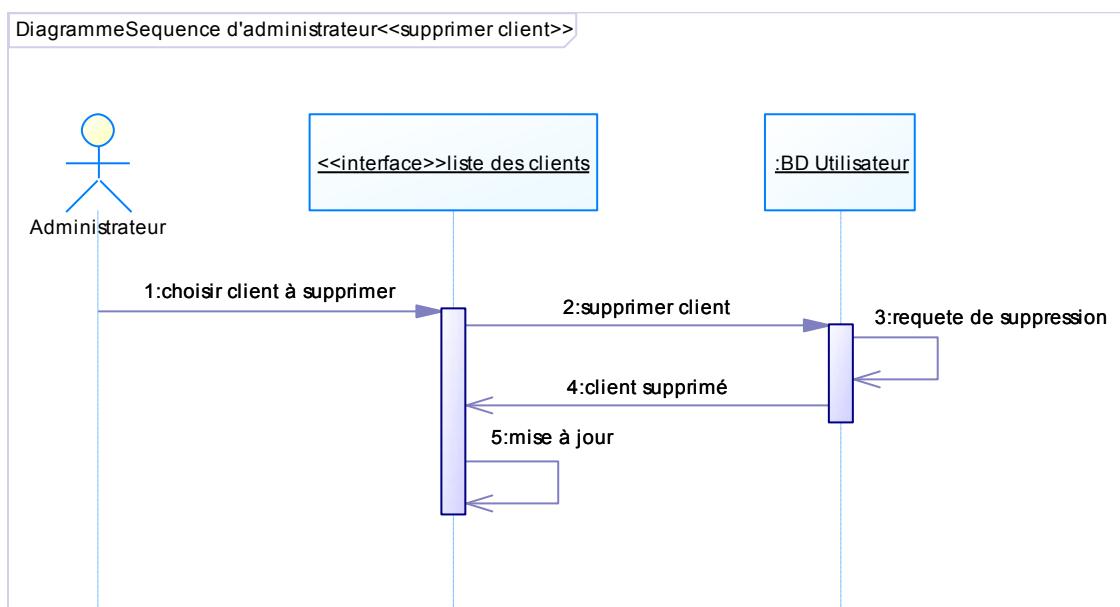


Figure : Diagramme de séquence d'administrateur « Supprimer client »

Pré condition :

L'administrateur doit être authentifié.

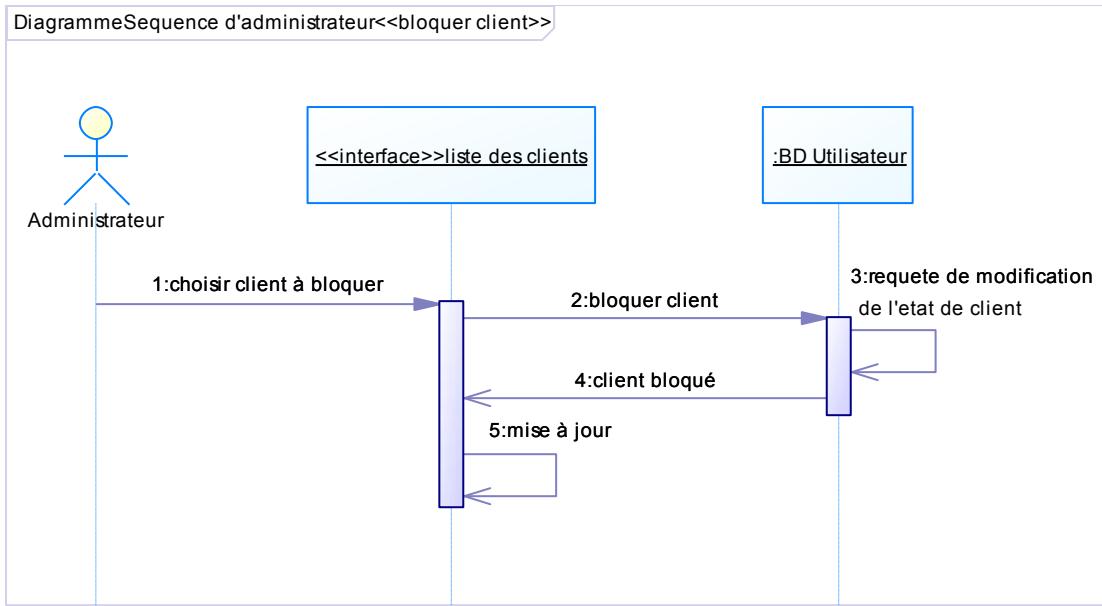
Scénario normal :

1. L'administrateur doit choisir le client à supprimer.
2. le système affiche un message pour la validation de suppression.
3. le système exécute la requête de suppression et supprime le client de la base de données.
4. le client est supprimé avec succès.
5. le mise à jour est effectuée à l'interface liste des clients.

Post condition :

Le client est supprimé.

❖ Diagramme de séquence d'administrateur « Bloquer client » :



❖ Figure : Diagramme de séquence d'administrateurs « Bloquer client »

Pré condition :

Le client doit être authentifié.

Scénario normal :

1. L'administrateur doit choisir le client à bloquer.
2. le système affiche un message pour la validation de blocage.
3. le système exécute la requête de blocage et bloque le client de la base de données.
4. le client est bloqué avec succès.
5. la mise à jour est effectuée à l'interface liste des clients.

Post condition :

Le client est bloqué.

❖ Diagramme de séquence « Ajouter événement au calendrier » :

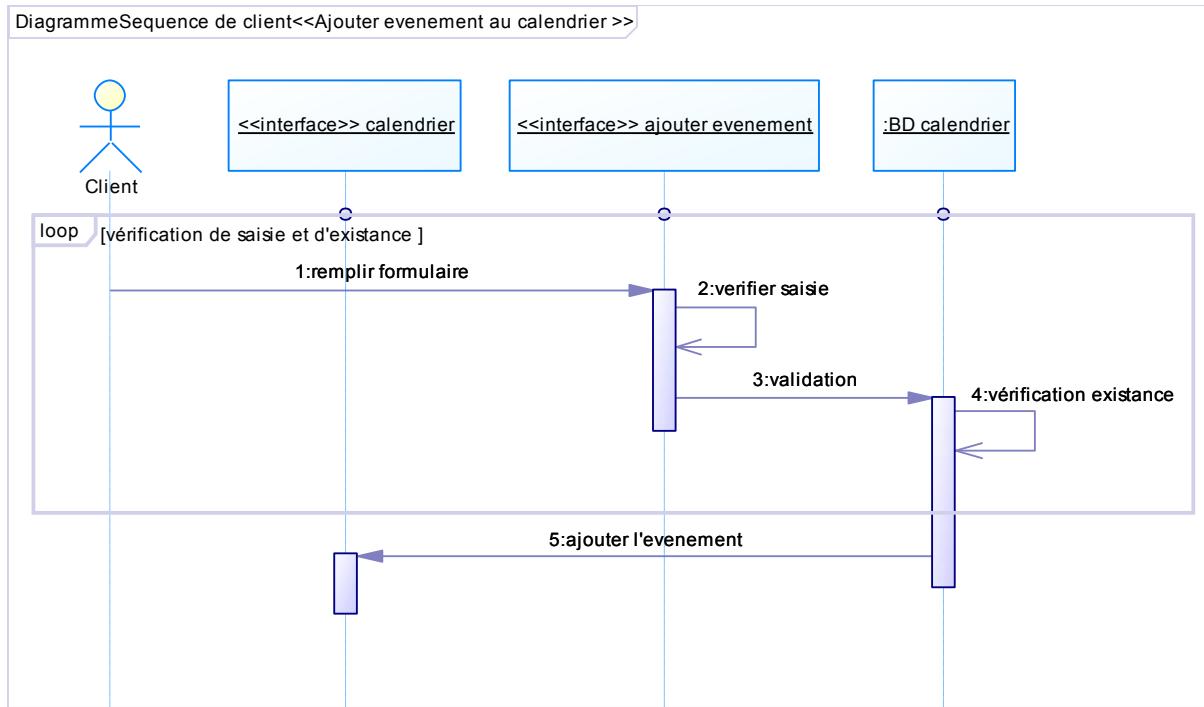


Figure : Diagramme de séquence de client « Ajouter événement au calendrier »

Pré condition :

L'opération d'ajout doit être demandée.

L'administrateur doit être authentifié.

Scénario normal :

1. Le client doit être saisi les informations de l'événement à ajouter.
2. le système contrôle les champs saisis.
3. le client valide la saisie.
4. le système vérifie l'existence de l'événement : Si les informations existent déjà, le système affiche un message «cet événement existe déjà ».
5. Si l'événement n'existe pas le système effectué l'ajout.

Post condition :

L'ajout d'événement est effectué.

❖ **Diagramme de séquence de client « Modifier événement »:**

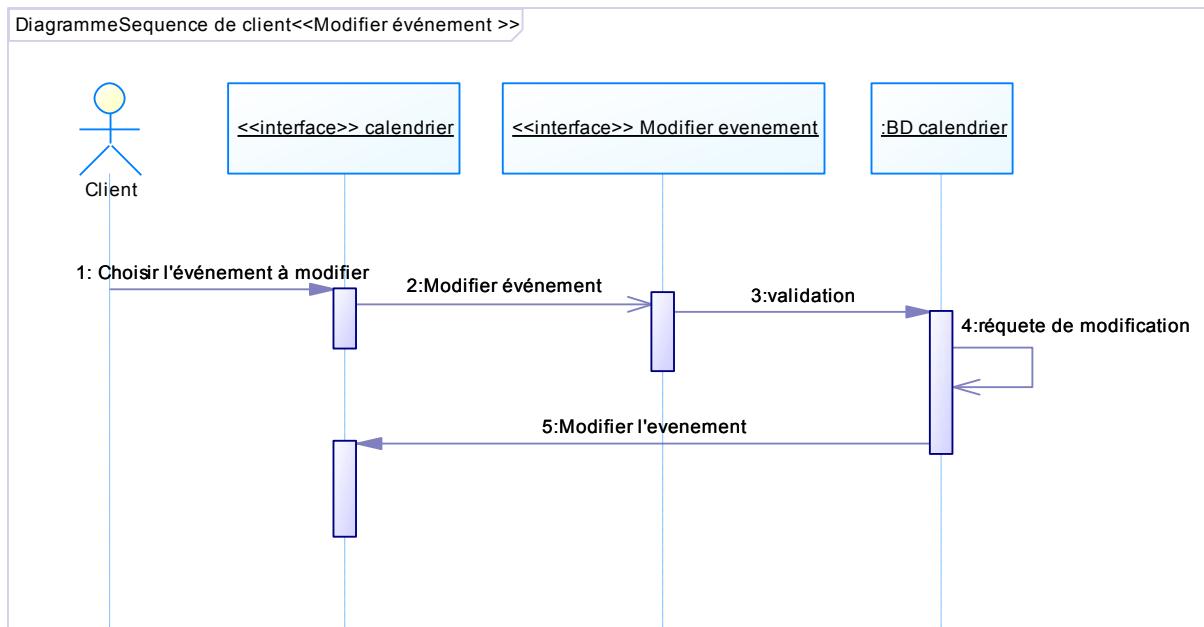


Figure : Diagramme de séquence de client « Modifier événement »

Pré condition :

Le client doit être authentifié.

Scénario normal :

- Le client choisit l'événement à modifier.
- Le système dirige l'interface de liste des événements vers modifier événement.
- Le client saisie les informations à modifier d'événement et valide le saisie.
- Le système exécute la requête de modification et enregistrer les informations modifiés.
- L'opération de modification d'événement est effectuée et sera affiché à la liste des événements.

Post condition :

L'événement est modifié.

❖ **Diagramme de séquence de client « Supprimer événement » :**

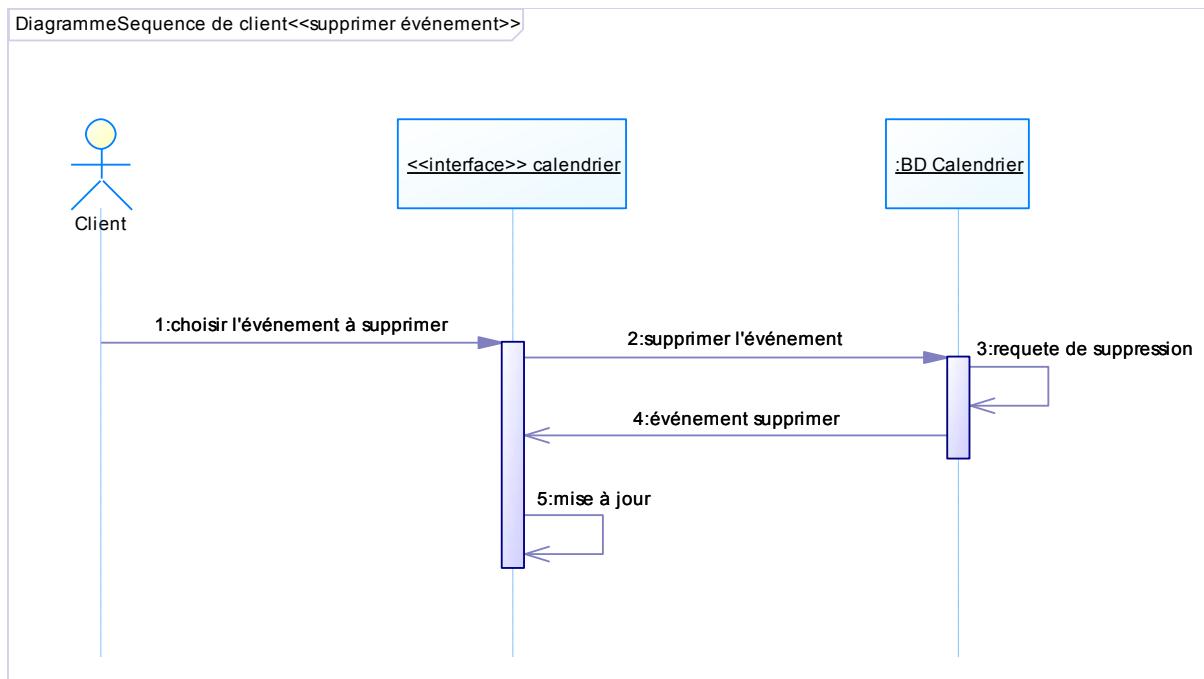


Figure : Diagramme de séquence de client « Supprimer événement »

Pré condition :

Le client doit être authentifié.

Scénario normal :

1. Le client doit choisir l'événement à supprimer.
2. le système affiche un message pour la validation de suppression.
3. le système exécute la requête de suppression et supprime l'événement de la base de données.
4. L'événement est supprimé avec succès.
5. le mise à jour est effectuée à l'interface liste des événements.

Post condition :

L'événement est supprimé.

❖ **Diagramme de séquence de client « Envoyer message contact »:**

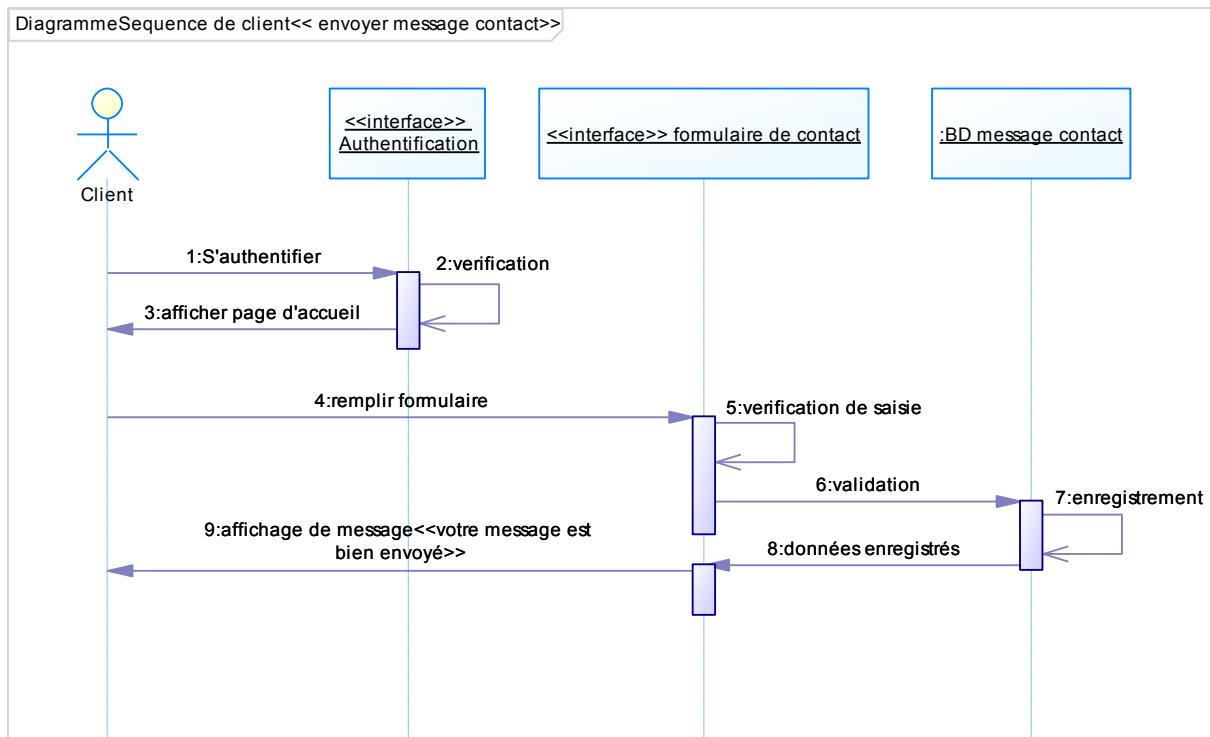


Figure : Diagramme de séquence de client « Envoyer message contact »

Pré condition :

Le client doit être authentifié.

Scénario normal :

1. Le client doit s'authentifier
2. le système effectué une vérification.
3. la page d'accueil client s'affiche pour le client
4. le client remplir le formulaire de message contact
5. le système effectué une vérification de saisie.
6. l'envoi de message contact est validé.
7. le message de contact est enregistré à la base de données
8. les données sont stockées à la base de données avec succès.
9. le système affiche un message pour le client « votre message est bien envoyé »

Post condition :

Le message contact est envoyé avec succès.

❖ **Diagramme de séquence de visiteur « Envoyer message contact » :**

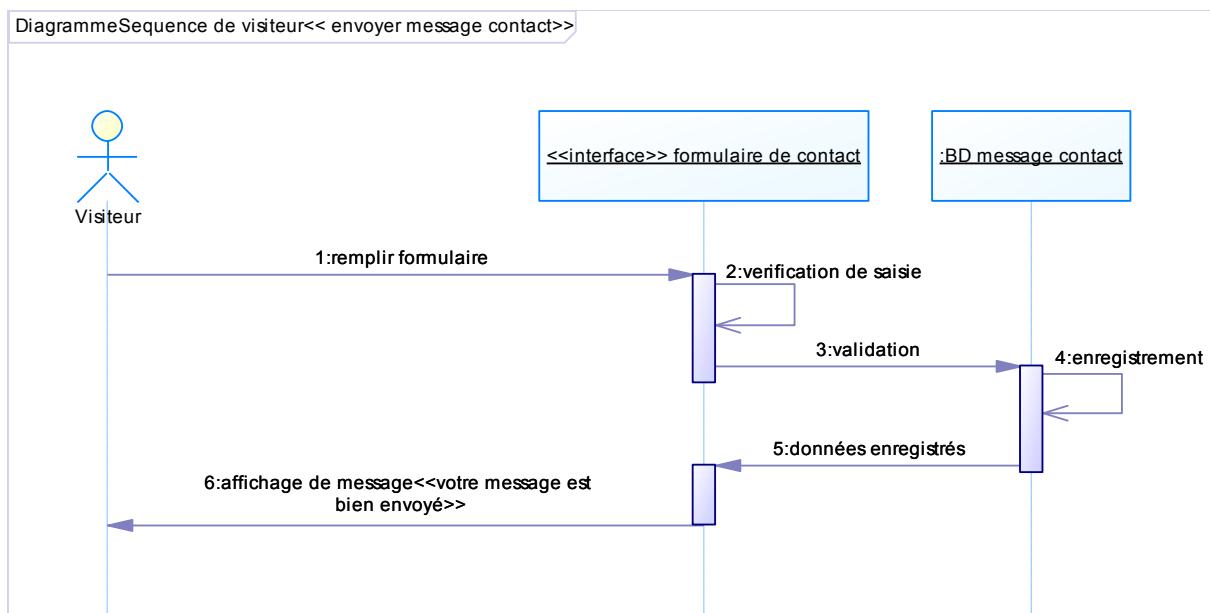


Figure : Diagramme de séquence de visiteur « Envoyer message contact »

Pré condition :

Le visiteur ne doit pas s'authentifié.

Scénario normal :

1. le visiteur remplir le formulaire de message contact
2. le système effectué une vérification de saisie.
3. l'envoi de message contact est validé.
4. le message de contact est enregistré à la base de données
5. les données sont stockées à la base de données avec succès.
6. le système affiche un message pour le visiteur « votre message est bien envoyé »

Post condition :

Le message contact est envoyé avec succès.

❖ **Diagramme de séquence d'administrateur « Supprimer message contact » :**

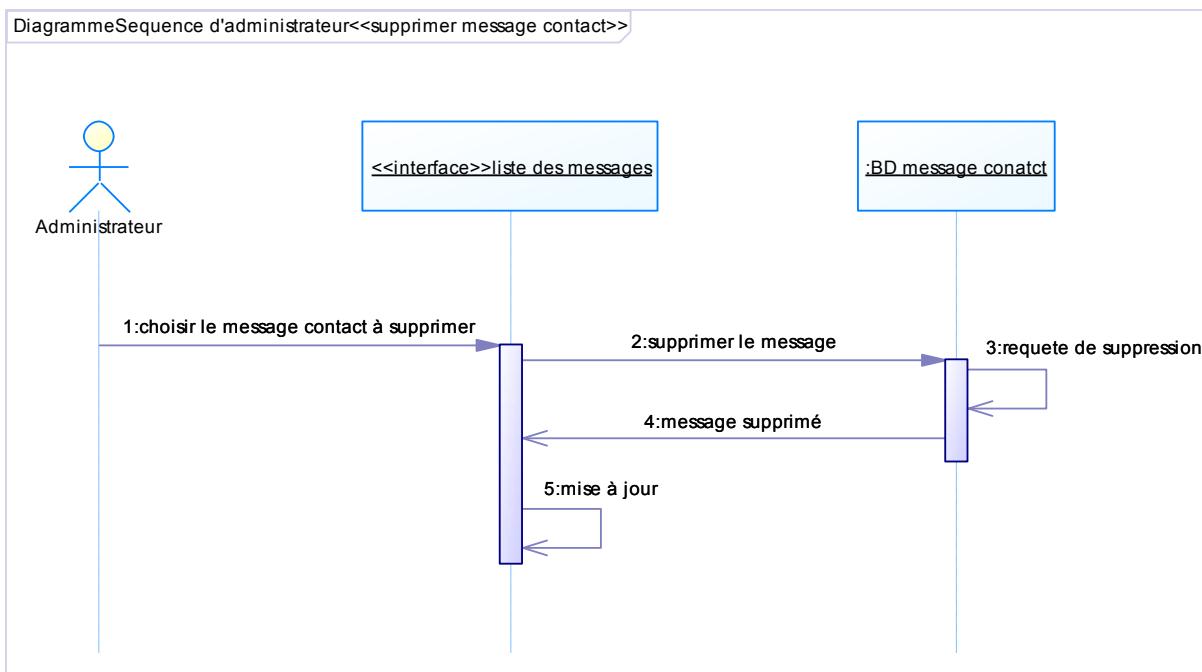


Figure : Diagramme de séquence d'administrateur « Supprimer message contact »

Pré condition :

L'administrateur doit être authentifié.

Scénario normal :

1. L'administrateur doit choisir le message à supprimer.
2. le système affiche un message pour la validation de suppression.
3. le système exécute la requête de suppression et supprime le message de la base de données.
4. le message est supprimé avec succès.
5. la mise à jour est effectuée à l'interface liste des messages.

Post condition :

Le message est supprimé.

C. Diagramme de collaboration

Le diagramme de collaboration fait partie de la vue dynamique. Il met l'accent sur les relations spatiales entre les objets. Les messages entre ces derniers peuvent être numérotés pour introduire la dimension temporelle (du point de vue ordre des messages).

❖ Diagrammes de classe d'analyse « Ajouter contact»

- Ajouter contact : C'est une classe interface permettant aux clients d'ajouter des nouveaux contacts.
- La fonctionnalité qu'offre cette interface est : Ajouter contact
- Contrôle Gestion de contact : C'est une classe contrôle qui permet de gérer toutes les opérations possibles sur les contacts
- Contact : C'est une classe entité contenant toutes les informations sur les contacts.

Figure : Diagramme de classe d'analyse « Ajouter contact »

Description textuelle :

- Pour ajouter un contact le client ouvre la fenêtre d'ajout des contacts.
- Le système affiche le formulaire d'ajout de contacts.
- Le client saisie les nouvelles informations du contact.
- Le système enregistre les nouvelles informations du contact.

❖ **Diagrammes de classe d'analyse « Modifier contact»**

- Modifier contact : C'est une classe interface permettant aux clients de modifier des contacts.
- La fonctionnalité qu'offre cette interface est : Modifier contact
- Contrôle Gestion de contact : C'est une classe contrôle qui permet de gérer toutes les opérations possibles sur les contacts
- Contact : C'est une classe entité contenant toutes les informations sur les contacts.

Figure18 : Diagramme de classe d'analyse « Modifier contact »

Description textuelle :

- Pour modifier un contact le client ouvre la fenêtre de la liste des contacts.
- Le système affiche la liste de contacts.
- Le client choisie le contact à modifier
- Le système affiche le formulaire de modification de contacts.
- Le client saisie les nouvelles informations du contact.
- Le système enregistre les nouvelles informations du contact.

❖ **Diagrammes de classe d'analyse « Supprimer contact»**

- Supprimer contact : C'est une classe interface permettant aux clients supprimer des contacts.

- La fonctionnalité qu'offre cette interface est : supprimer contact
- Contrôle Gestion de contact : C'est une classe contrôle qui permet de gérer toutes les opérations possibles sur les contacts
- Contact : C'est une classe entité contenant toutes les informations sur les contacts.

Figure18 : Diagramme de classe d'analyse « Supprimer contact »

Description textuelle :

- Pour supprimer un contact le client ouvre la fenêtre de la liste des contacts.
- Le système affiche la liste de contacts.
- Le client choisie le contact à supprimer
- Le système affiche une fenêtre pour la validité de suppression de contacts.
- Le système supprime les informations du contact choisi.

D.Diagramme d'activité

Ce diagramme se dérivent des diagrammes d'état-transition, il décrit le comportement d'une opération ou d'une méthode, il permet la mise en évidence de l'organisation des activités. Il représente à la fois les comportements conditionnels et parallèles et regroupe graphiquement les activités par objet.

A un niveau macroscopique, les diagrammes d'activité permettent de décrire des enchaînements de fonctionnalités. Ils complètent donc bien les cas d'utilisation au niveau de l'analyse des besoins.

A un niveau microscopique, les diagrammes d'activité permettent, par exemple, de décrire l'algorithme d'une action d'un diagramme d'états

❖ Diagramme d'activité « Authentification »

Le système affiche le formulaire d'authentification et l'utilisateur saisit son nom d'utilisateur et mot de passe, le système fait une vérification, s'ils sont validés le système affiche la page d'accueil, si non il affiche un message d'erreur.

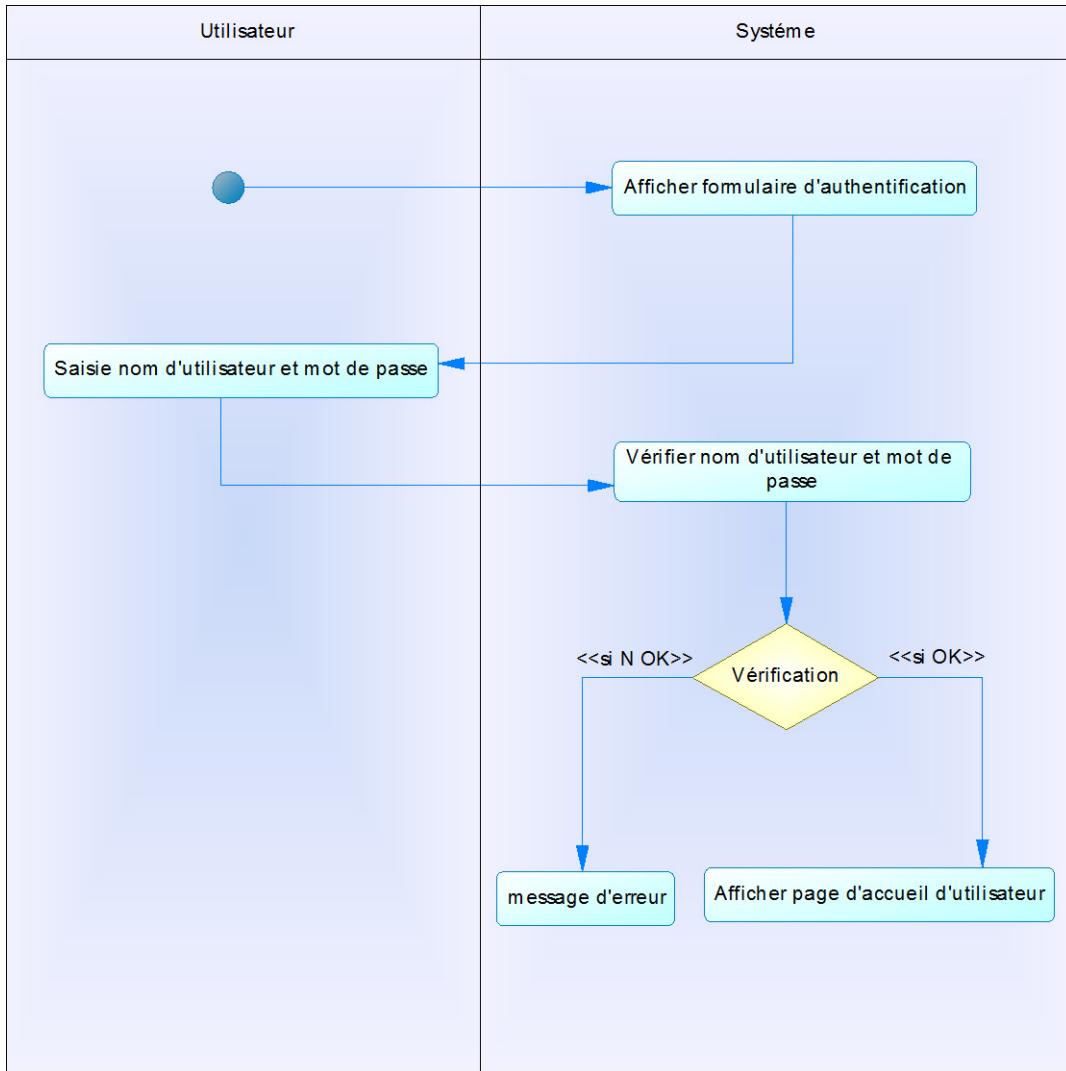


Figure25: Diagramme d'activité « Authentification »

❖ Diagramme d'activité « Ajouter contact »

Le client envoie une demande d'ajout d'un contact et remplir la formulaire, une vérification sera faite, si elles sont validées l'ajout s'effectue sinon le système affiche un message d'erreur.

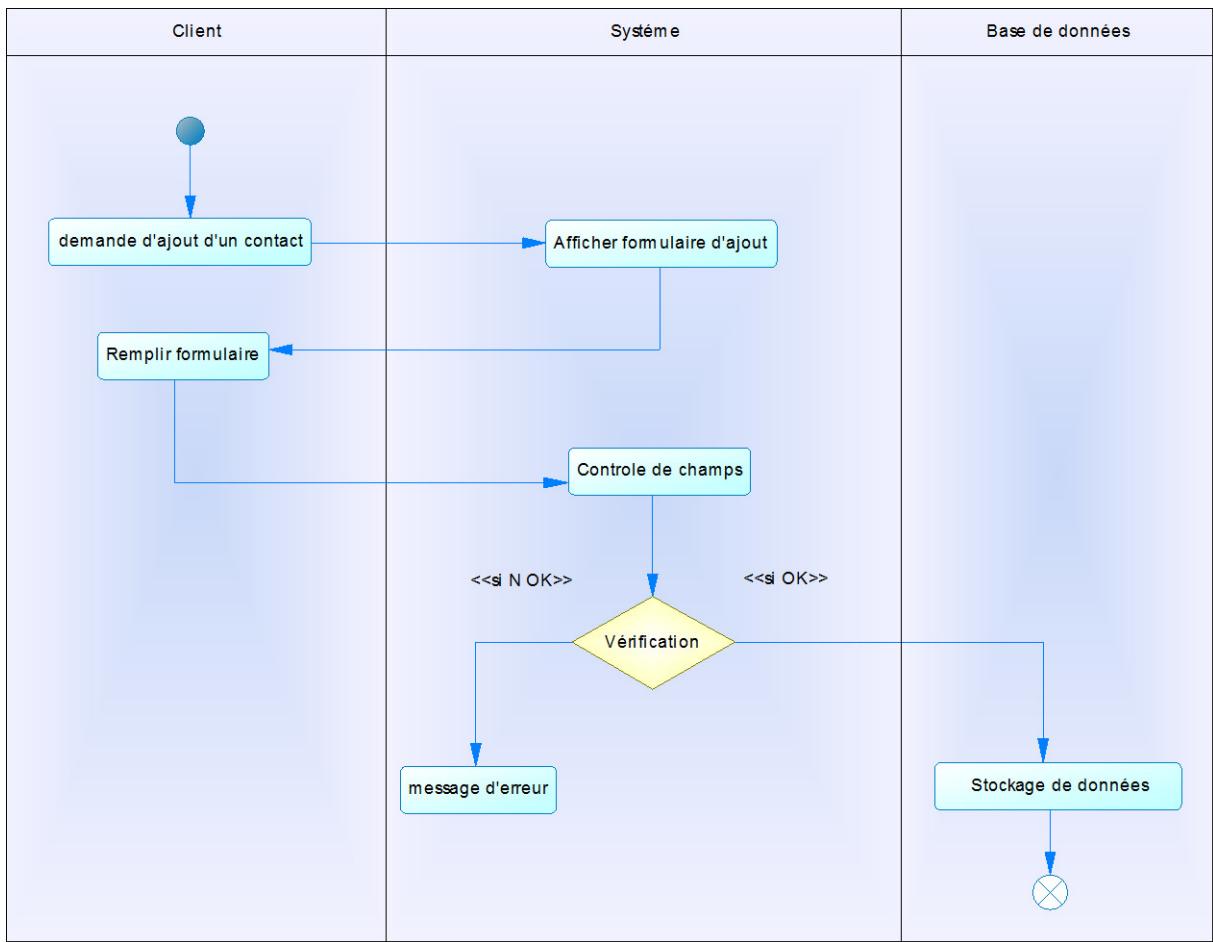


Figure25: Diagramme d'activité « Ajouter contact »

❖ Diagramme d'activité « Modifier contact »

Le client envoie une demande de modification d'un contact, la liste des contacts s'affiche, le client sélectionne le contact à modifier, il saisie les informations à modifier au formulaire de modification, une vérification sera faite, si elles sont validées la modification s'effectue, sinon le système affiche un message d'erreur.

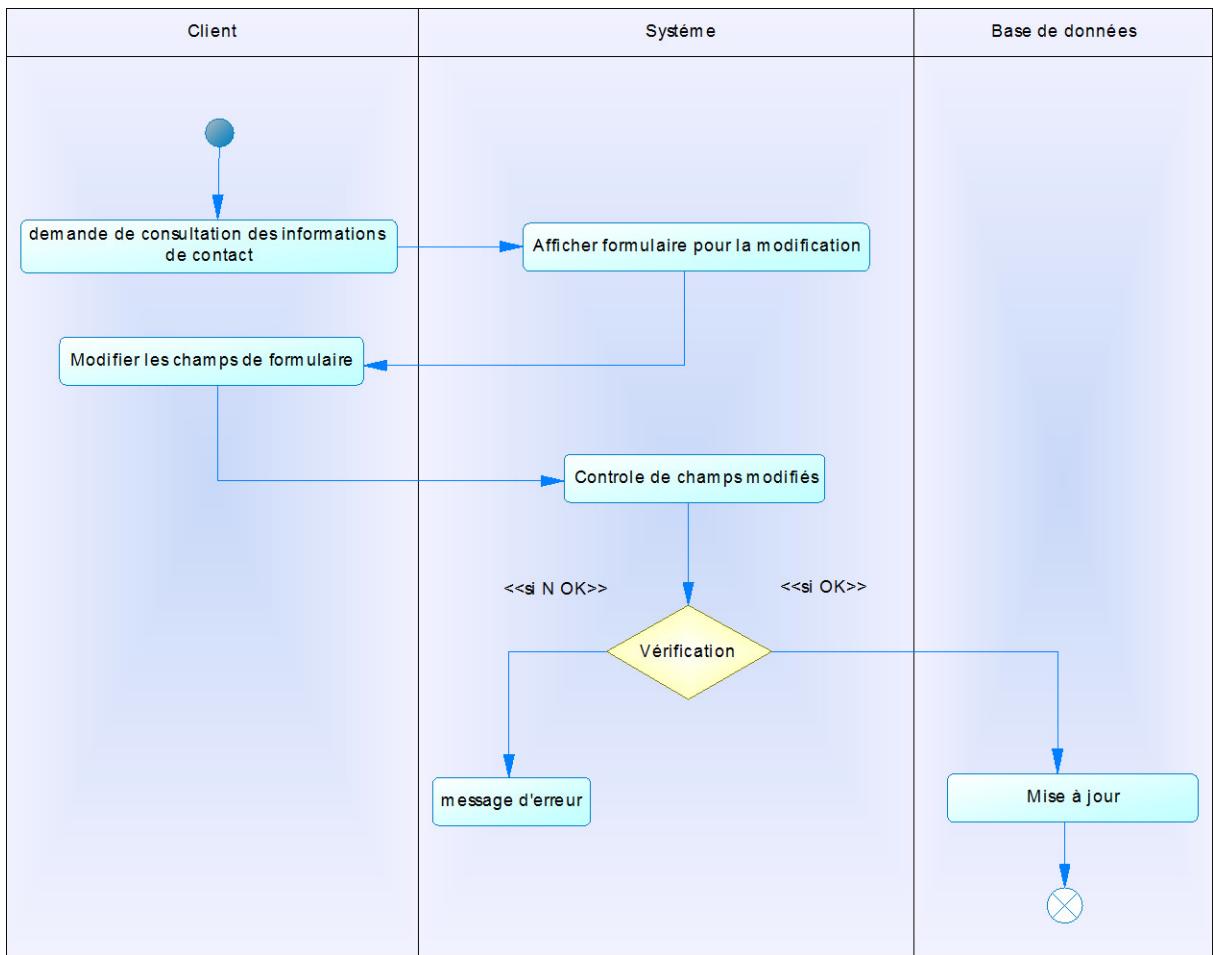


Figure25: Diagramme d'activité « Modifier contact »

❖ Diagramme d'activité « supprimer contact »

Le client envoie une demande de suppression d'un contact, la liste des contacts s'affiche, le client sélectionne le contact à supprimer puis valide, la suppression s'effectue avec succès.

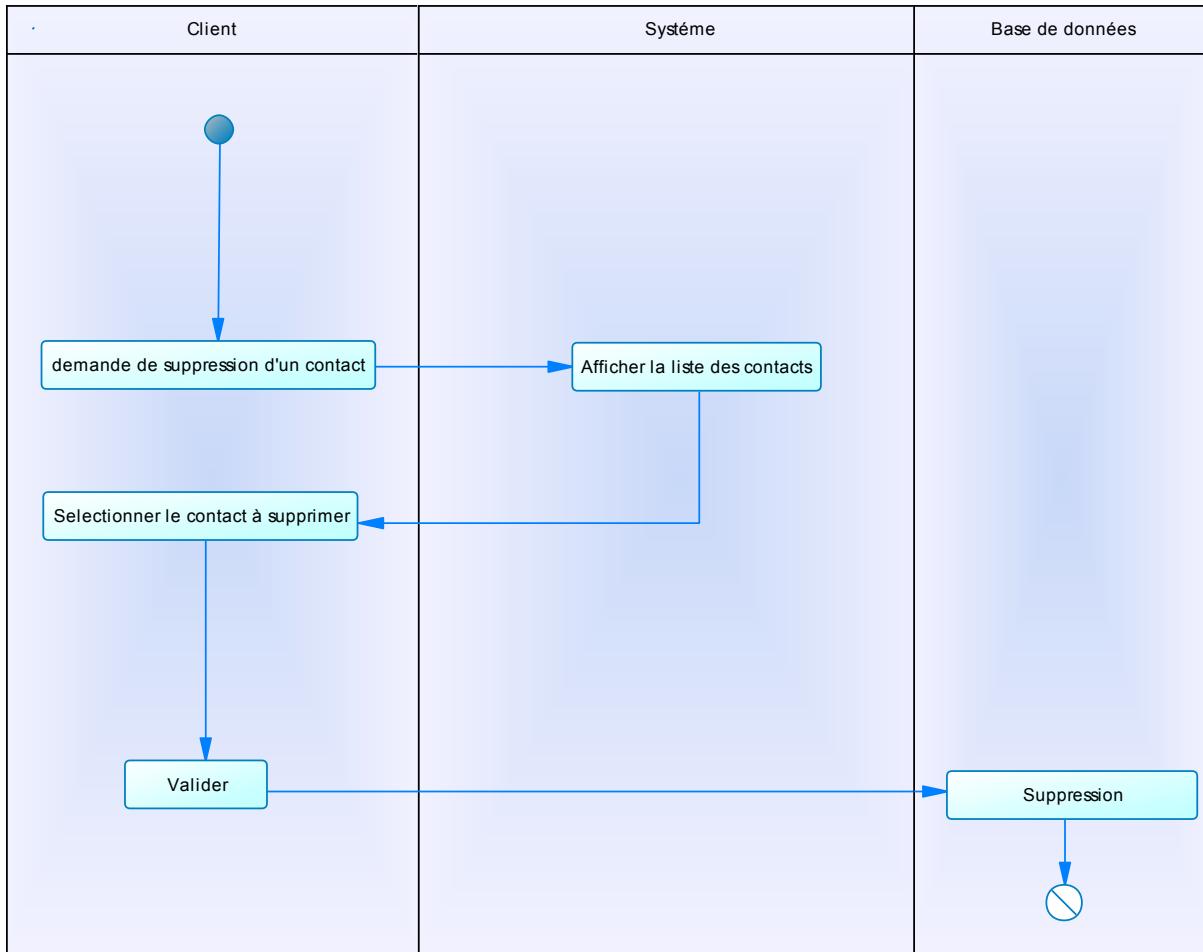


Figure25: Diagramme d'activité « Supprimer contact »

E. Diagramme de déploiement :

Le diagramme de déploiement spécifie un ensemble de constructions qui peut être utilisé pour définir l'architecture d'exécution de systèmes qui représentent l'affectation d'artefacts logiciels à des nœuds. Les nœuds sont connectés via des chemins de communication pour créer des systèmes de réseau d'une complexité quelconque. Les nœuds sont en général définis d'une manière imbriquée et représentent soit des périphériques matériels, soit des environnements d'exécution de logiciels. Les artefacts représentent des éléments concrets du monde physique qui sont le résultat d'un processus de développement.

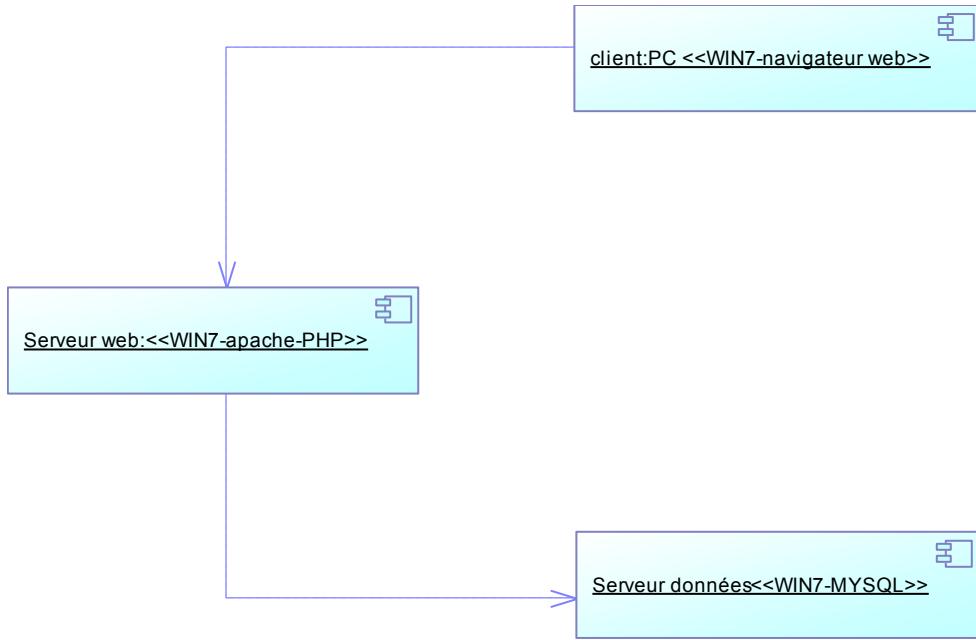


Figure : Diagramme de déploiement

F. Diagramme de classe

Le diagramme de classe est une collection d'éléments du modèle (statique), tels que des choses, des interfaces et leurs relations, connectées entre eux comme un graphe. Un diagramme de classe montre uniquement les aspects statiques du modèle.

Le diagramme suivant présente les différents classes et relations utiles pour la réalisation qui constitue la base de données de notre application (voir dictionnaire des données : Annexe).

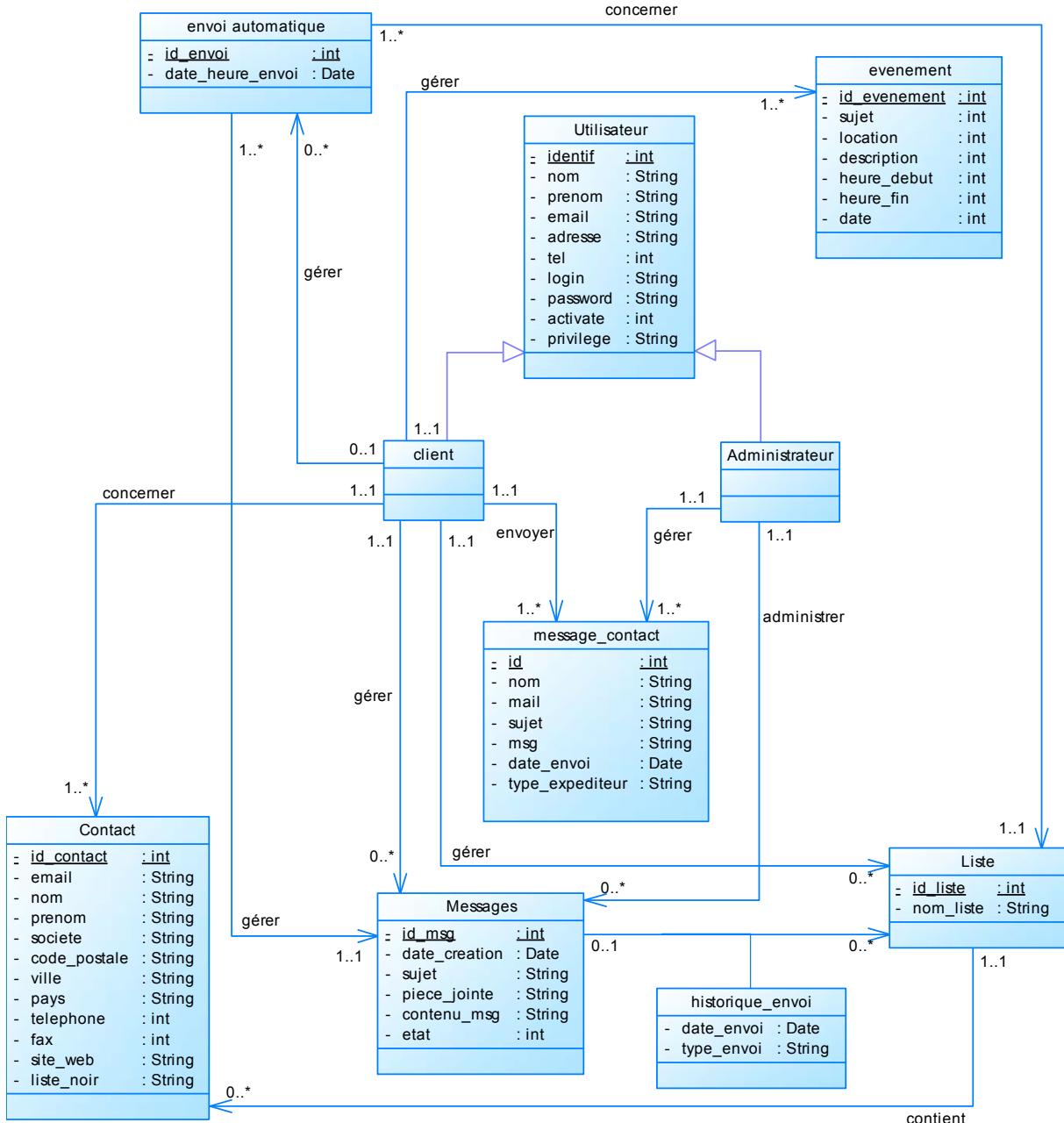


Figure : Diagramme de classe

IV .Conclusion

Dans ce chapitre j'ai présenté mon application Web par les différents modèles et diagrammes pour donner une idée globale pour rendre notre projet plus claire et compréhensive sur les différentes cotés. Toutes ces taches vont être traduites sur des interfaces qu'on va représenter dans le chapitre suivant.

[Tapez le titre du document]

I. Introduction

La phase de réalisation met en valeur les interfaces graphiques de l'application. Donc nous allons commencer tout d'abord par l'identification des langages et des outils de développement. Puis nous allons présenter les différentes phases d'implémentation et quelques interfaces de notre application web pour décrire leurs fonctionnements.

II. Langages et outils de développement :

1. HTML :



Le langage utilisé pour la conception d'une page web est Hyper Text Markup Langage. Il ne s'agit pas de langage de programmation au sens propre, mais d'un simple langage de description d'une page web. Le fichier qui contient la description de cette page porte en générale l'extension .html ou .htm .Il est constitué du texte et des lignes aux images à afficher, repartie entre les balises (par exemple : <p>....</p>) qui détermine la façon dont ces éléments seront présentés dans le navigateur. Certaines de ces balises permettent aussi de transformer un texte ou une image en un lien hypertexte (<a>.....). Ces hyperliens sont des éléments très important dans une page web, puisqu'ils permettent d'organiser la navigation en reliant les pages entre elles. Ainsi l'internaute peut passer d'une page à l'autre par un simple click sur ces liens. D'où l'expression de naviguer sur le web.

2. CSS :



CSS (Cacading Style Sheets :feuilles de style en cascade) sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web consortium(W3c).

Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien supportés par les navigateurs web dans les années 2000. Les css ont plusieurs avantages.

La structure du document et la présentation peuvent être gérées dans des fichiers séparés.

La conception d'un document se fait dans un premier temps sans se soucier de la présentation, ce qui permet d'être plus efficace.

Dans le cas d'une application web, la présentation est uniformisée : les documents (page HTML) font référence aux mêmes feuilles de style. Cette caractéristique permet de plus une remise en forme rapide de l'aspect visuel.

3. Adobe photoshop CS5 :



Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par adobe. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d'images ex nihilo. Photoshop est un logiciel dit <<bitmap>>. Reconnu aussi par les infographistes professionnels à travers sa puissante galerie de filtres et d'outils graphiques performants, et utilisé par une grande majorité des studios et agences de créations.

4. Adobe dreamweaver CS5 :



Adobe dreamweaver est un éditeur de type “tel écrit tel ecran” , cette excellente formule remplaçant désormais dans la terminologie informatique le sigle anglophone WYSIWYG (What You See Is What You Get : ce que vous voyez est ce que vous obtiendrez)

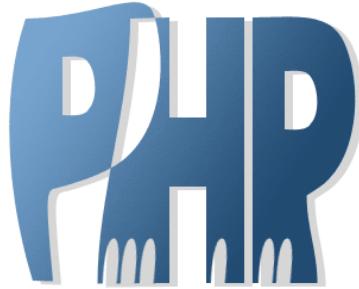
Le logiciel Dreamweaver CS5 produites par Adobe donne aux concepteurs et développeurs, débutants et professionnels, la capacité à créer des normes des sites web ou d’application web avec un air de confiance en soi. Les propriétaires de ce nouveau logiciel est en mesure de concevoir visuellement ou dans le code. Voici certaines des caractéristiques que l’CS5 Dreamweaver a:

Intelligent assistance de codage – Vous serez capable d’écrire un code plus propre plus rapidement que jamais. Vous aurez sûrement l’idée de profiter de conseils de code avec JavaScript, HTML et les frameworks Ajax comme le prototype, Spry et jQuery.

Amélioration globale de soutien CSS – On vous donnera des outils puissants CSS qui permettent de construire votre application web. Vous serez également en mesure de montrer visuellement le modèle de boîte CSS et pas besoin d’utilitaires séparés. Il n’y aura pas besoin de modifier manuellement tout code CSS.

Soutien des principales technologies – Vous aurez la certitude que vous serez capable de concevoir et de développer votre application Web dans un environnement qui est favorable à peu près toutes les technologies Web de pointe. Certains de ces derniers incluent les éléments suivants: HTML, CSS, Ajax, Adobe ColdFusion logiciels, XHTML, XML, JavaScript, PHP et ASP.

2.1.PHP (Personal Home Page) :



PHP (Hypertext Preprocessor) est un langage de script qui est principalement utilisé pour être exécuté par un serveur HTTP, mais il peut fonctionner comme n'importe quel langage interprété en utilisant les scripts et son interpréteur sur un ordinateur. On désigne parfois PHP comme une plate-forme plus qu'un simple langage.

Sa syntaxe et sa construction ressemblent à celles des langages C++, C et Java à la différence que le PHP peut être directement intégré dans du code HTML.

Ses principaux atouts sont :

- ✓ La gratuité et la disponibilité du code source.
- ✓ La simplicité d'écriture de scripts.
- ✓ La possibilité d'inclure le script PHP au sein d'une page HTML.
- ✓ La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plates-formes : Unix, Linux, Windows ...).
- ✓ L'intégration au sein de nombreux serveurs web (Apache,...)

Le langage PHP est un langage extrêmement puissant : il permet de créer des pages web, au travers lesquelles l'utilisateur peut échanger des informations avec le serveur ; c'est ce qu'on appelle des pages web dynamiques. Programmer en PHP est assez simple.

En revanche, PHP n'est pas un langage compilé, c'est un langage interprété par le serveur : le serveur lit le code PHP, le transforme et génère la page HTML. Pour fonctionner, il a donc besoin d'un serveur web. Donc si vous souhaitez utiliser des pages en PHP dans votre site web, pour les tester, il faudra les exécuter sur un serveur web.

2.2.Fonctionnement de PHP :

PHP n'est pas un langage compilé, c'est un langage interprété par le serveur : le serveur lit le code PHP, le transforme et génère la page HTML. Pour fonctionner, il a donc besoin d'un serveur web. De ce fait une plateforme minimale de base pour l'exécution d'un site web développé en PHP comprend :

- ◆ L'interpréteur PHP (serveur PHP)
- ◆ un serveur web (EasyPHP, Apache, IIS, ...)

Lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête à un serveur HTTP. Si la page contient du code PHP, l'interpréteur PHP du serveur le traite et renvoie du code généré (HTML).

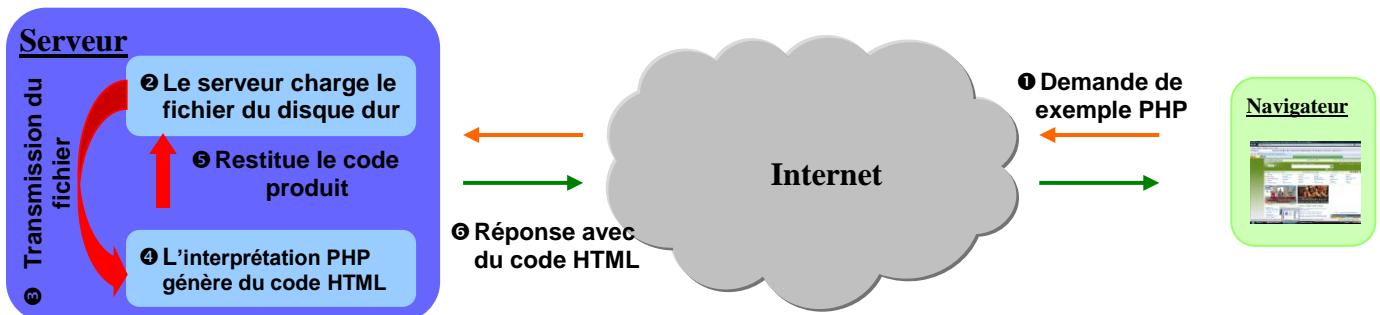


Figure : L'interprétation PHP génère du code HTML

De ce fait le code PHP n'est jamais visible sur la page finale consultée par le client. Ainsi en éditant la source de la page on n'y trouvera que du code HTML.

2.3. Avantages et Limitations de langage PHP:



➤ Les avantages:

Le langage PHP possède les mêmes fonctionnalités que les autres langages permettant d'écrire des scripts CGI, comme collecter des données, générer dynamiquement des pages web ou bien envoyer et recevoir des cookies,

- ✓ La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données et la simplicité d'interfaçage avec eux.
- ✓ La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL).
- ✓ La simplicité d'écriture de scripts (apprentissage rapide).
- ✓ La possibilité d'inclure le script PHP au sein d'une page HTML.
- ✓ La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plates-formes : Unix, Linux, Windows ...).
- ✓ PHP est utilisable sur la majorité des systèmes d'exploitation, comme Linux, de nombreuses variantes Unix (incluant HP-UX, Solaris et OpenBSD), Microsoft Windows, Mac OS X, RISC OS et d'autres encore.

L'intégration au sein de nombreux serveurs web actuels : Apache, Microsoft Internet Information Server, Personal Web Server, Netscape et iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPD et beaucoup d'autres encore.

➤ **Limitations :**

- ✓ Langage interprété
- ✓ L'orienté objet reste limité
- ✓ Pas adéquat en terme de rapidité et de maintenabilité pour des projets de grandes envergures

2.4.WAMP SERVER:



Wampserver est un paquetage WAMP, une plateforme de développement web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Wampserver n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur WAMP SERVER web Apache et un serveur de base de données MySQL), un interpréteur de script PHP, ainsi qu'une administration SQL PhpMyAdmin. Il dispose d'interfaces d'administration permettant de gérer les alias (dossier virtuels disponibles sous Apache), et le démarrage /arrêt des serveurs. Il permet donc d'installer en une seule fois tous

les nécessaires au développement local du PHP. Par défaut le serveur Apache crée un nom de domaine virtuel (car sur local) http://127.0.0.1 ou http://localhost .

2.5.PHPMyAdmin :



L'un des outils le plus utiles sur ce serveur est PHPMyAmin.

PHPMyAdmin permet de gérer votre base de données MYSQL en fournissant une interface graphique pour l'insertion, la modification et la suppression des données.

2.6.Apache :



Apache est le serveur web le plus répondu permettant à des clients d'accéder à des pages web, c'est-à-dire en réalité des fichiers en format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant.

Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.

2.7.MYSQL :



MYSQL est un véritable serveur de base de données SQL Multiutilisateur et Multi-threaded ce qui signifie qu'il peut utiliser plusieurs CPU. IL permet de créer et de gérer des bases de données relationnelles.

La nature de ce SGBD a permis son extension pour devenir la première base de données dans le monde –même si MYSQL est encore en développement, il propose déjà un ensemble de fonctionnalités riches et extrêmement utiles sous Linux et Windows.

Pour la réalisation de notre base nous avons utilisé MySQL grâce à sa fiabilité et sa disponibilité gratuitement sous une licence libre et il possède une interface intuitive et conviviale avec PHP et il est aussi très suffisant pour une base de données tel que celle qu'on va la réaliser.

2.8.JQuery



jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant AJAX) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.

Le framework contient notamment les fonctionnalités suivantes :

- Parcours et modification du DOM (y compris le support des sélecteurs CSS 1 à 3 et un support basique de XPath) ;
- Événements ;
- Effets et animations ;
- Manipulations des feuilles de style en cascade (ajout/suppression des classes, d'attributs...) ;
- AJAX ;
- Plugins ;
- Utilitaires (version du navigateur...).

III. Réalisation :

A. « Planning » de réalisation :

planning de réalisation	01/02	15/02	01/03	15/03	01/04	15/04	01/05
Création de la base de données	X						
Création des maquettes	X	X					
Création des scripts PHP de base pour la partie administrateur		X	X				
Création des scripts PHP de base pour la partie client			X	X	X		
Création des scripts de contrôle de champs						X	
Création des scripts d'authentification et de sécurité					X	X	
Saisie de données réelles							X
Création d'une galerie Jquery à l'Intro							X
test d'application		X	X	X	X	X	X

Tableau 1: Le « Planing » de réalisation

B. Etape réalisation du système :

✚ Interface d'accueil de l'application:

C'est une introduction à notre application pour présenter les différents services offerts.

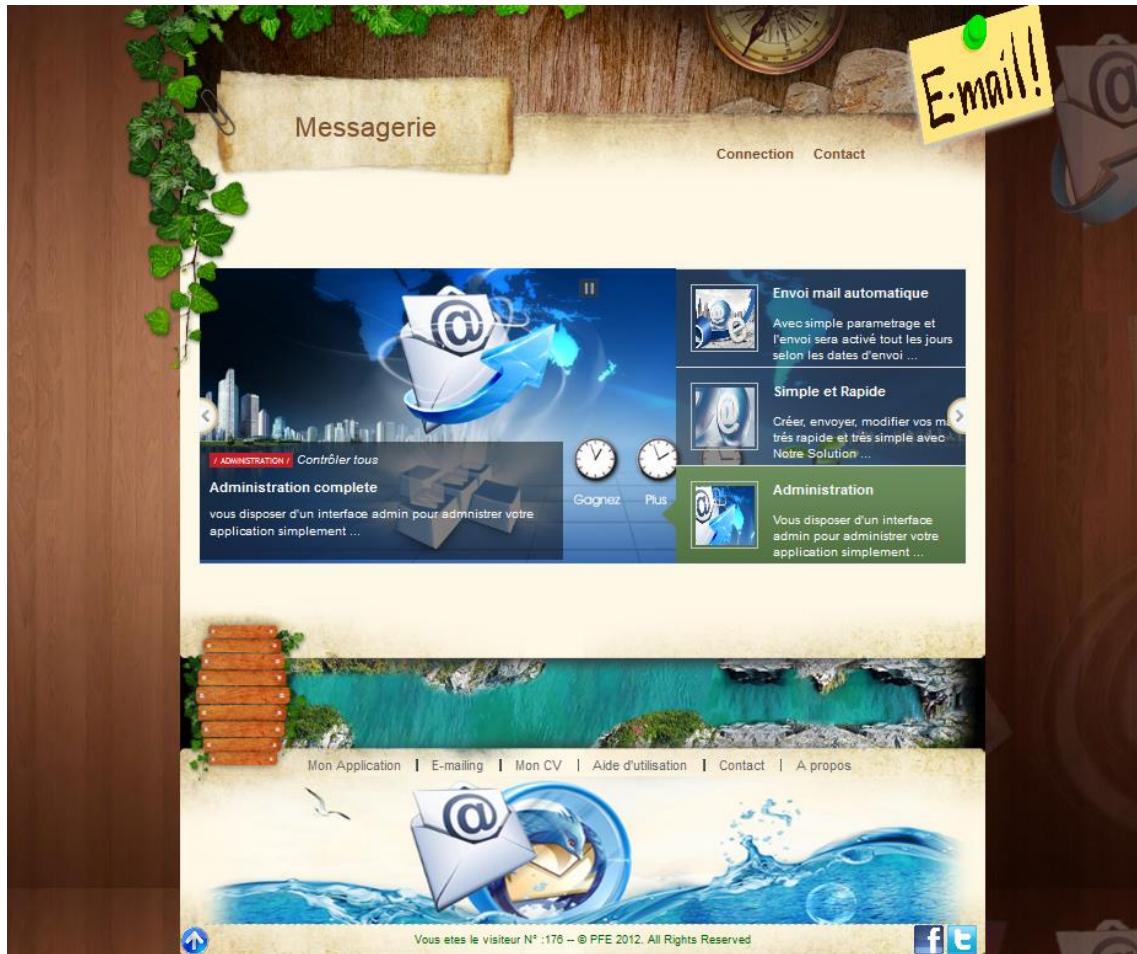


Figure: interface d'accueil

1. Les interfaces coté client :

✚ Interface d'inscription:

L'interface ci-dessous présente le formulaire d'inscription client, dans le cas où le client oublie de saisir les champs obligatoires, un message d'erreur s'affiche pour la vérification.

The screenshot shows a registration form titled "Ajouter Utilisateur". The form fields include:

- Nom*: Bellazreg
- Prenom: Riadh
- Email*: riag1991@hotmail.com
- Adresse: Moknine
- Tél: 73698569
- Nom d'utilisateur*: riadh
- Mot de passe*: [REDACTED]
- Confirmer mot de passe*: [REDACTED]

A red error bubble points to the password fields with the message: "* Votre champs n'est pas identique". Below the password fields is a CAPTCHA image with the text "wbC4l5".

Buttons at the bottom include "S'inscrire" and "Cliquer sur l'image pour changer le texte".

Figure: interface d'inscription

Interface d'authentification :

La figure ci-dessous représente la page d'authentification de l'administrateur. En effet, L'authentification est la première tâche qu'un administrateur doit effectuer pour accéder au menu administrateur approprié.

Dans le cas où il saisit des informations erronées un message d'erreur lui sera affiché à fin de corriger les informations : son nom d'utilisateur ou mot de passe.

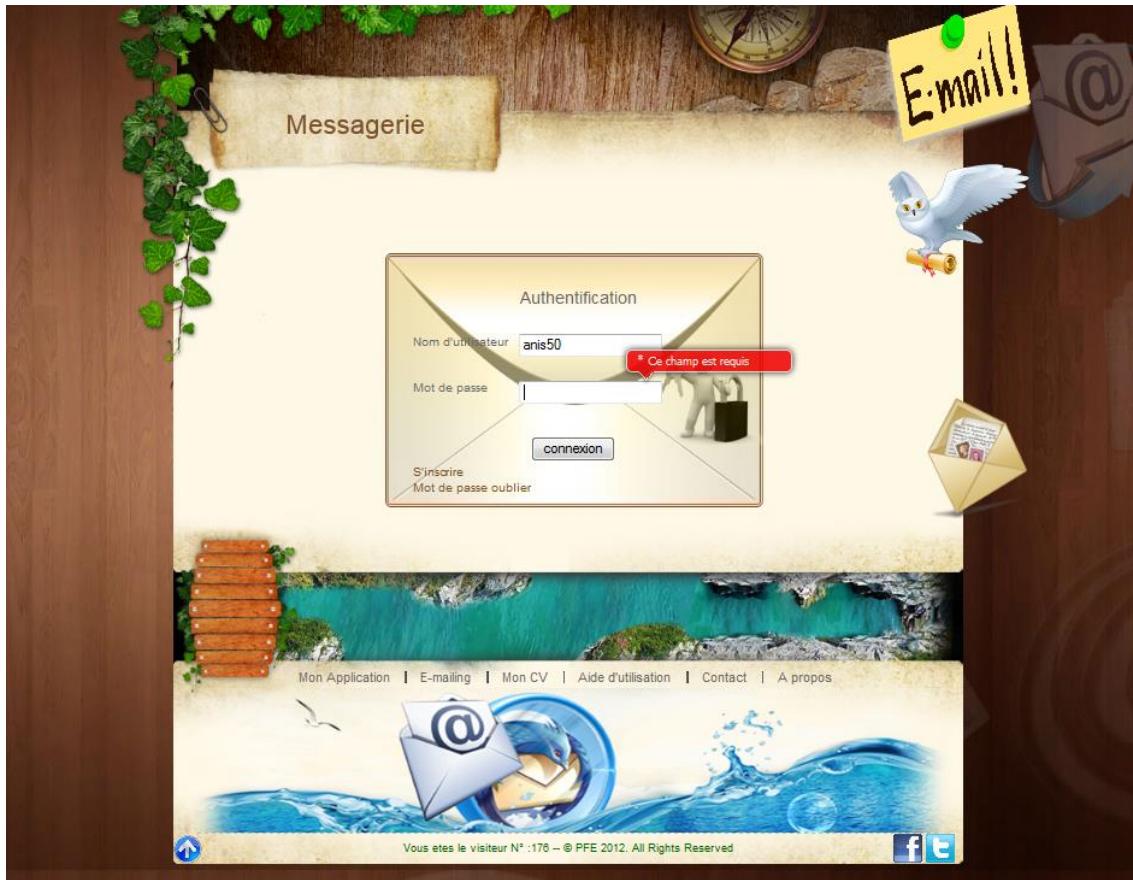


Figure: interface d'authentification utilisateur

 **Interface d'accueil espace client :**

La page d'accueil espace client contient un menu principal composé des tâches suivantes:

- ✓ Authentification
- ✓ Composer messages
- ✓ Envoyer messages
- ✓ Gestion des messages.
- ✓ Gérer l'envoi automatique des messages
- ✓ Gestion des contacts.
- ✓ Importer/Exporter des contacts.
- ✓ Gérer les listes de contact.
- ✓ Protéger données d'utilisateur.
- ✓ Consulter les statiques d'envoie.
- ✓ Analyser avec Google Analytics.
- ✓ Gestion des événements de calendrier.
- ✓ Contacter l'administrateur



Bienvenue Dans Notre Application du Messagerie.

Figure: interface Menu principale de client

Interface gérer listes des contacts :

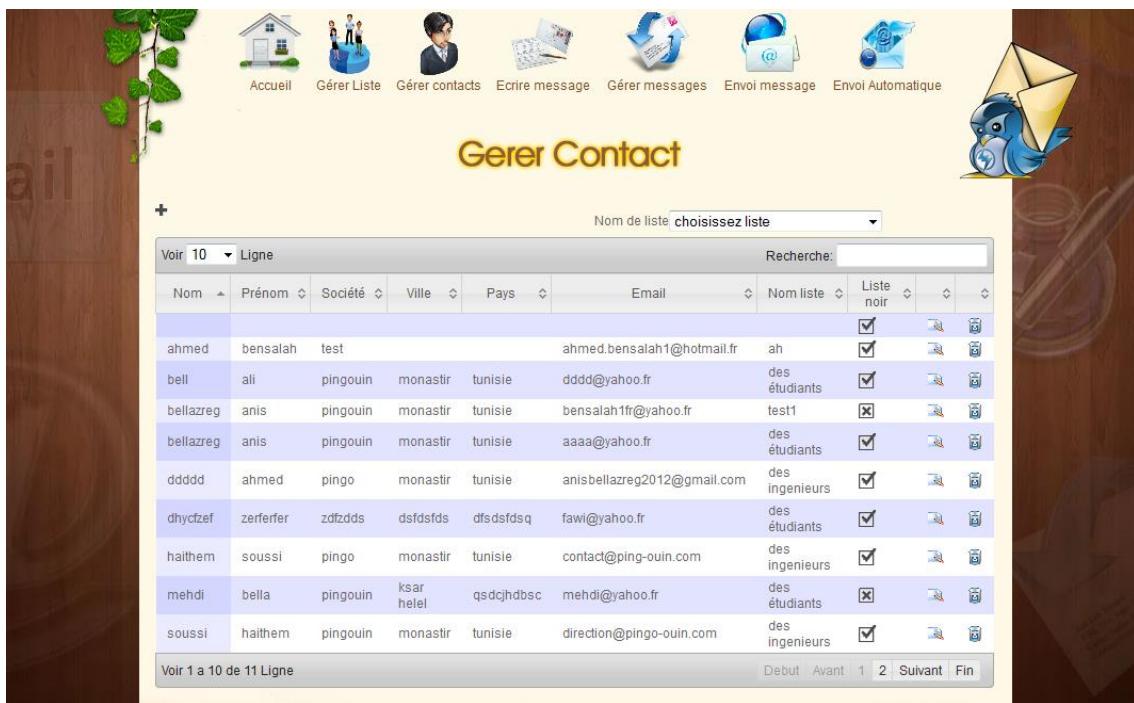
Cette interface nous présente la liste des listes de contacts ainsi les différents opérations que le client peut faire telles que l'ajout, la modification, la suppression des listes.

Nom
ah
des ingenieurs
des étudiants
jnijnjkn
qs dqd
test1

Figure: interface Gérer listes de contacts

 **Interface gérer les contacts :**

Cette interface nous présente les contacts ainsi les différents opérations que le client peut faire telles que l'ajout, la modification, la suppression des contacts ou l'ajout d'un contact à la liste noir.



The screenshot shows a software interface titled "Gérer Contact". At the top, there is a navigation bar with icons for Accueil, Gérer Liste, Gérer contacts, Écrire message, Gérer messages, Envoi message, and Envoi Automatique. Below the navigation bar, the title "Gérer Contact" is displayed in yellow. A search bar labeled "Recherche:" is present. The main area is a table listing 11 contacts. The columns are: Nom, Prénom, Société, Ville, Pays, Email, Nom liste, and Liste noir. Each row contains a set of icons for actions like edit, delete, and export. The "Liste noir" column indicates if a contact is in the black list, with a checked checkbox for most entries. The table footer shows "Voir 1 à 10 de 11 Ligne" and navigation links for Debut, Avant, Suivant, and Fin.

Nom	Prénom	Société	Ville	Pays	Email	Nom liste	Liste noir	Actions		
ahmed	bensalah	test			ahmed.bensalah1@hotmail.fr	ah	<input checked="" type="checkbox"/>			
bell	ali	pingouin	monastir	tunisie	dddd@yahoo.fr	des étudiants	<input checked="" type="checkbox"/>			
bellazreg	anis	pingouin	monastir	tunisie	bensalah1fr@yahoo.fr	test1	<input type="checkbox"/>			
bellazreg	anis	pingouin	monastir	tunisie	aaaa@yahoo.fr	des étudiants	<input checked="" type="checkbox"/>			
ddddd	ahmed	pingo	monastir	tunisie	anisbellazreg2012@gmail.com	des ingénieurs	<input checked="" type="checkbox"/>			
dhyczef	zerferfer	zdfzdds	dsfdsfsds	dfsdsfdsq	fawi@yahoo.fr	des étudiants	<input checked="" type="checkbox"/>			
haithem	soussi	pingo	monastir	tunisie	contact@ping-ouin.com	des ingénieurs	<input checked="" type="checkbox"/>			
mehdi	bella	pingouin	ksar helel	qsdcjhdbsc	mehdi@yahoo.fr	des étudiants	<input type="checkbox"/>			
soussi	haithem	pingouin	monastir	tunisie	direction@pingo-ouin.com	des ingénieurs	<input checked="" type="checkbox"/>			

Figure: interface Gérer les contacts

 **Interface ajouter contacts :**

A partir de cette interface, le client peut ajouter des contacts.



Figure: interface Ajouter un nouveau contact

 **Interface Modification des coordonnées d'un contact:**

A partir de l'interface ci-dessous, le client peut modifier les cordonnées des contacts.

Figure: modifier des contacts

Interface de suppression d'un contact :

A partir de cette interface, le client peut supprimer un contact.

Figure: suppression des contacts

Interface importer/exporter des clients:

A partir de cette interface, le client peut importer des contacts d'un fichier Excel vers la base de données ou exporter une liste des contacts de notre base de données vers un fichier Excel vers.



Figure: importation d'un contact

Interface composer messages:

A partir de cette interface, le client peut créer un message dans le but de l'envoyer au même temps ou de sauvegarder pour l'envoyer plus tard.

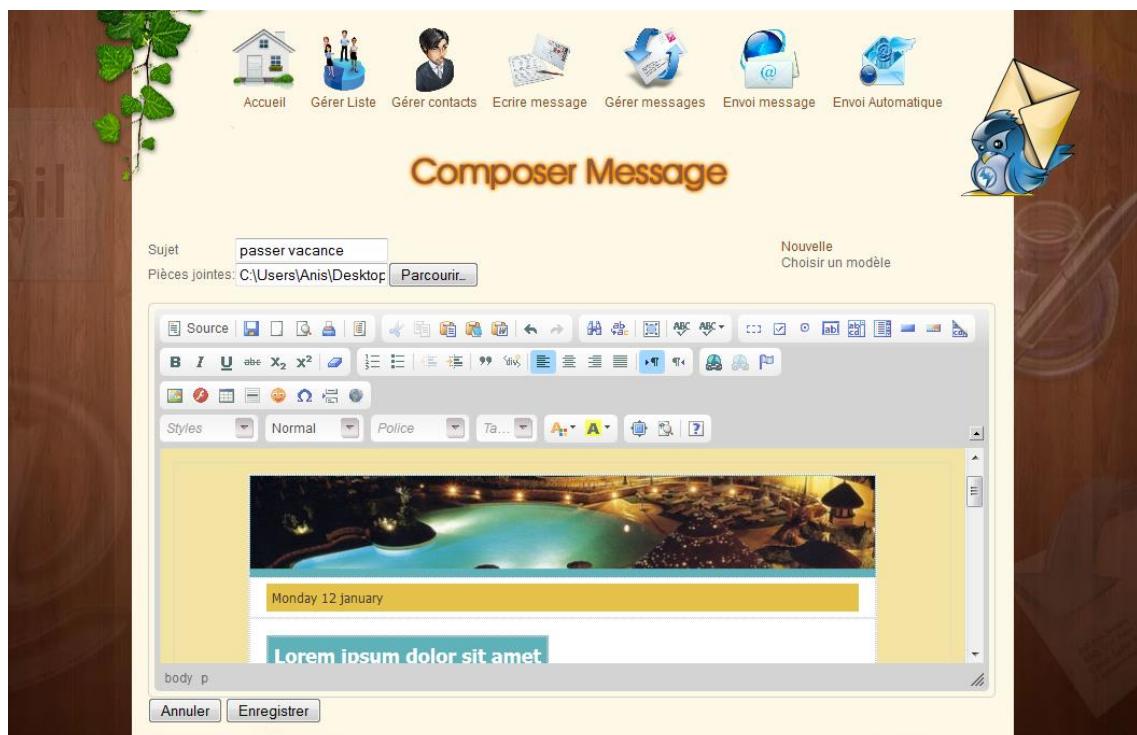


Figure: composer un message

Interface envoyer messages: (dynamique)

A partir de cette interface, le client peut envoyer un message à une liste des contacts.



Figure 1: envoyer un message

Interface envoi automatique des messages: (dynamique)

A partir de cette interface, le client peut envoyer un message à une liste des contacts à un jour et une heure bien défini pour effectuer l'envoi automatique.



Figure: envoi automatique d'un message

Interface gérer messages:

Cette interface nous présente les messages ainsi les différents opérations que le client peut faire telles que la modification, la suppression des messages.

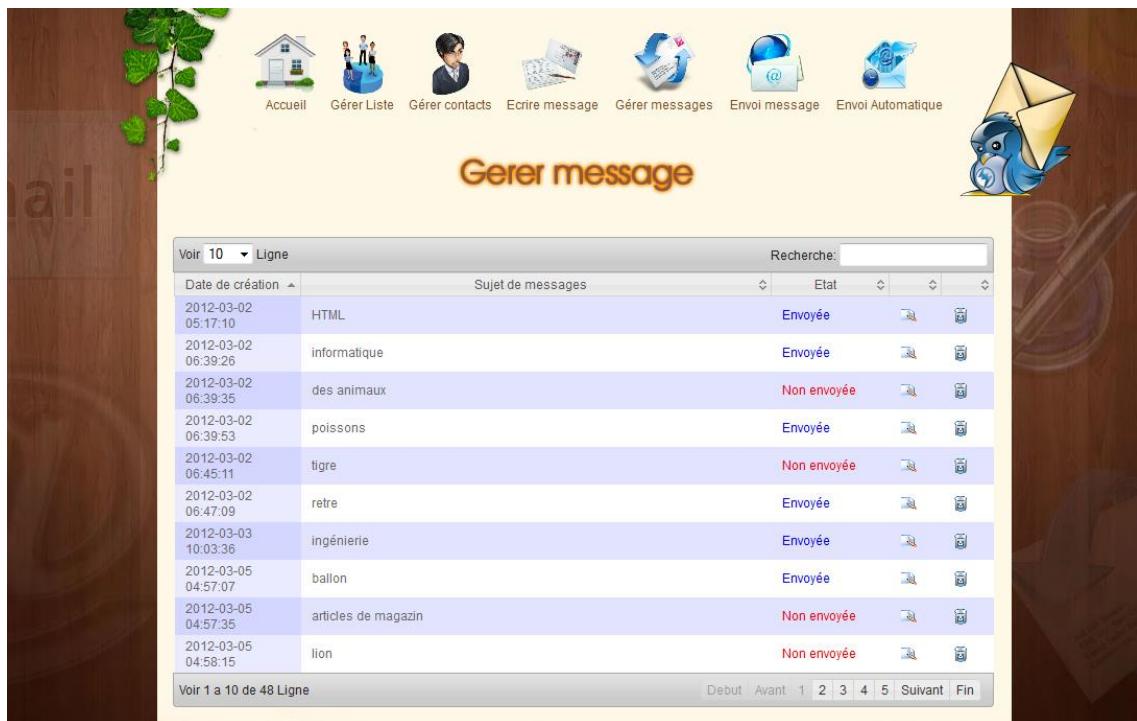


Figure: Interface gérer messages

⊕ Interface protéger vos données:

L’interface ci-dessous permet au client la modification de leur mot de passe pour protéger leurs données.



Figure: Interface protéger vos données

⊕ Interface historique d’envoi:

Cette interface permet au client de récupérer et savoir le type d'envoi des messages soit manuel, soit automatique selon listes de contacts.

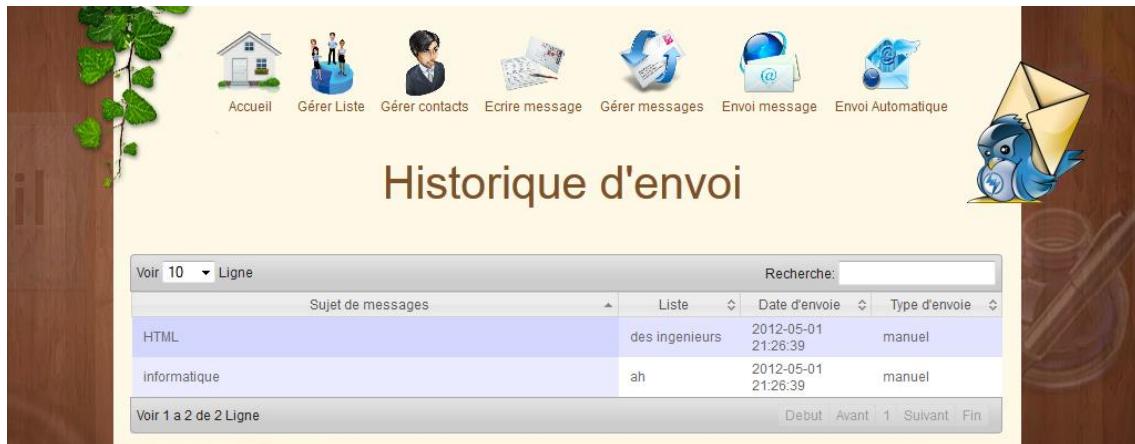


Figure: Interface historique d'envoi

⌚ Interface gestion de calendrier :

Cette interface présente le calendrier ainsi les différents opérations que le client peut faire telles que l'ajout, la modification, la suppression des événements.

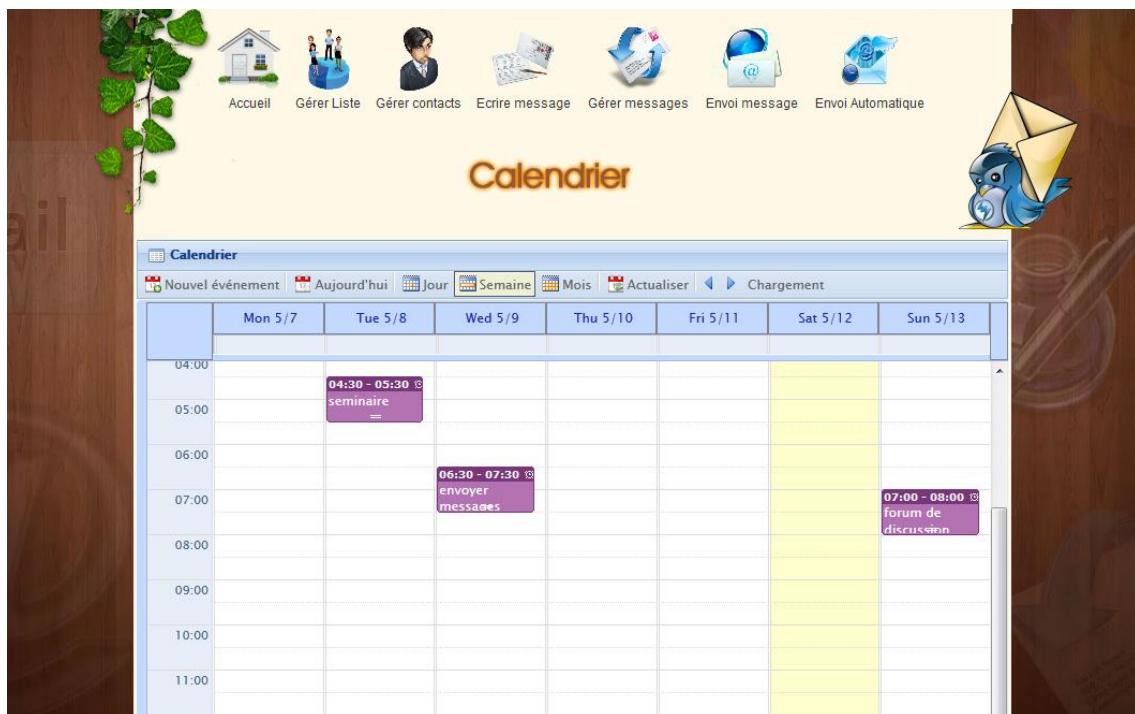


Figure: Interface calendrier

Interface statistique d'envoi:

L'interface ci-dessous présente les statistiques d'envoi des messages selon l'état d'envoi des messages envoyés et des messages non envoyés.

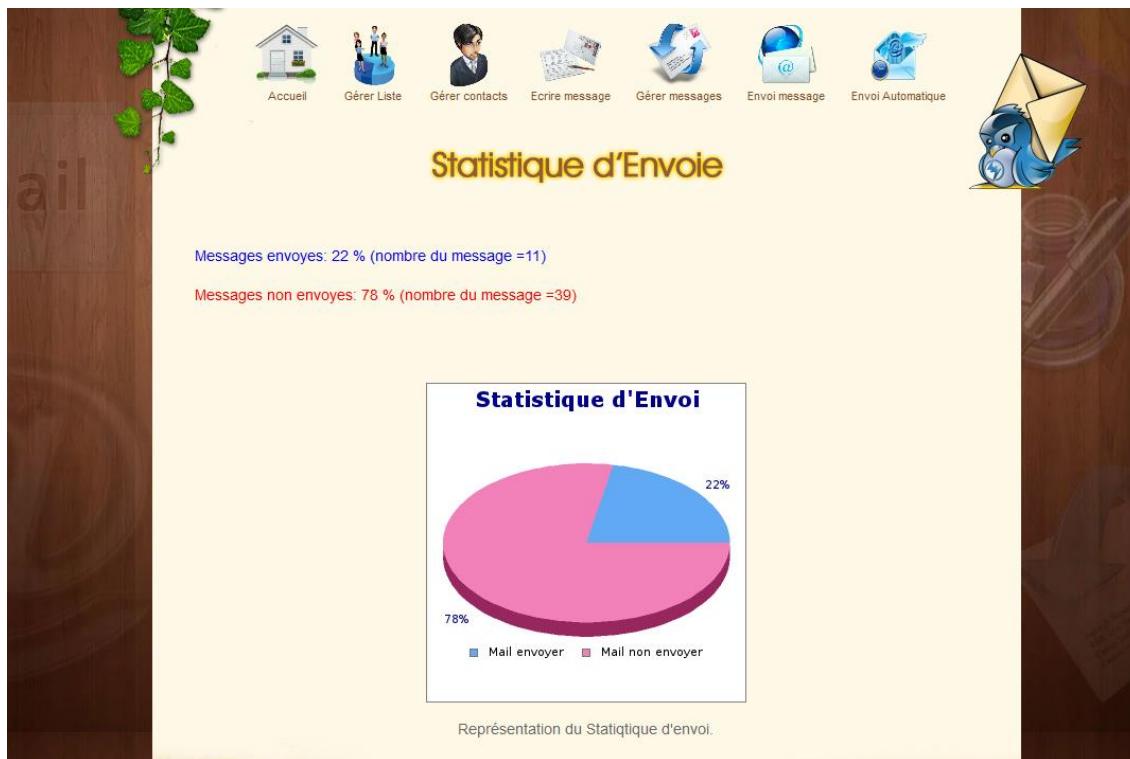


Figure: Interface statistique d'envoi

Interface formulaire de contact:

Cette interface présente le formulaire contact qui permet au client de contacter l'administrateur de l'application.

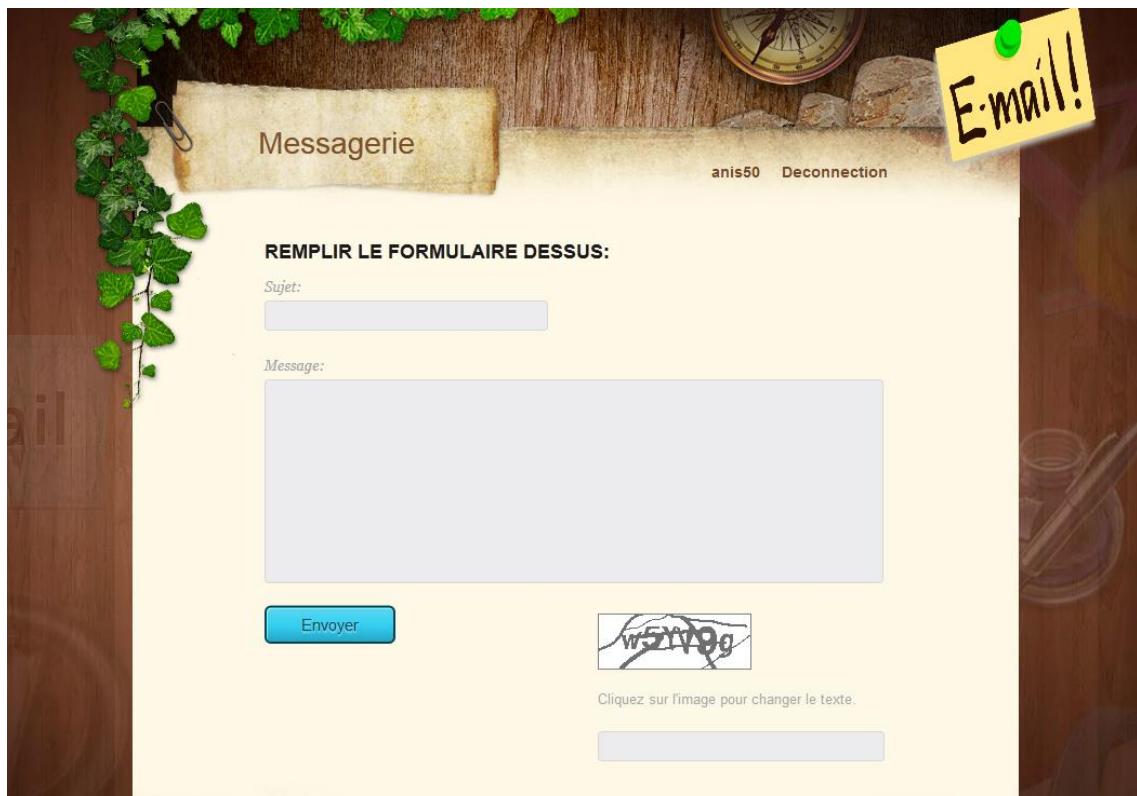


Figure: Interface formulaire de contact

1. les interfaces coté administrateur

NB : Tous les pages de la partie administrateur sont protégés contre les intrusions.

L'administrateur avoir :

- ✓ la possibilité d'ajout, de modification, de suppression et de blockage des clients.
- ✓ La Réception des messages de contacts et répondre ou supprimer des messages.
- ✓ La Consultation des messages créent par les clients et il a le droit de suppression des messages indésirable.
- ✓ La consultation de l'historique d'envoi de messages.
- ✓ La consultation des statistiques à Google analytics.

❖ **Interface menu administrateur :** (page statique)

Le menu principal permet à l'administrateur de consulter les différentes rubriques de l'application.

Elle affiche les liens vers toutes les pages de l'espace administrateur tout en garantissant la sécurité des pages.

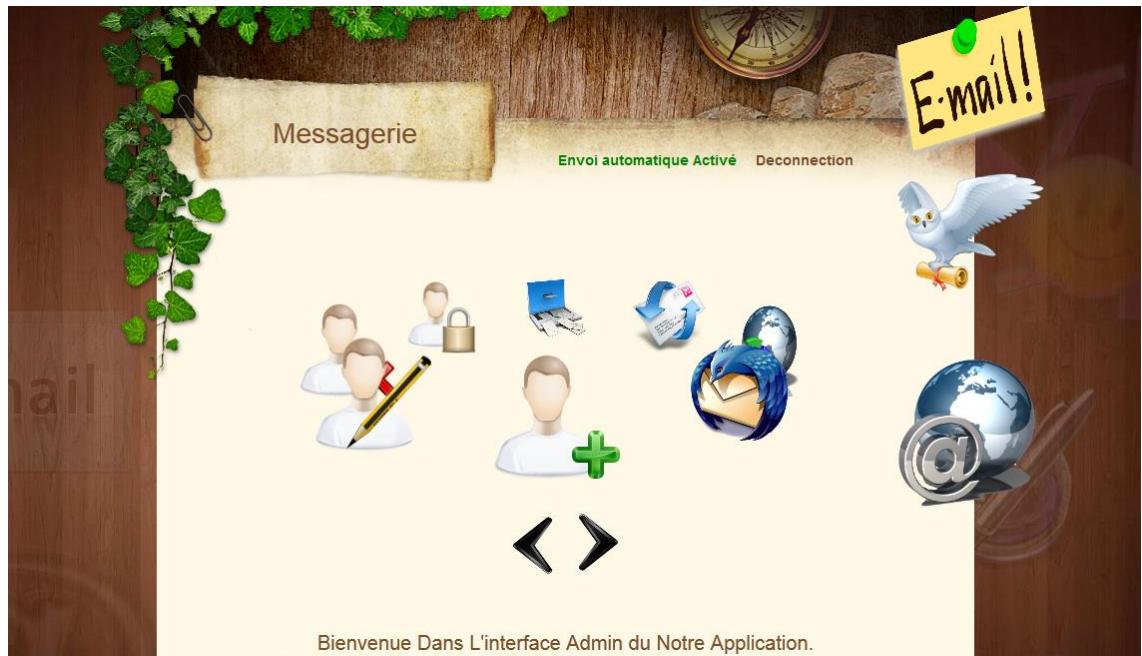


Figure: menu principale de l'administrateur

❖ **Interface Ajouter un nouveau client:**

Cette interface permet à l'administrateur de l'application d'ajouter un nouveau client

Ajouter Utilisateur

Nom	<input type="text"/>
Prenom	<input type="text"/>
Email*	<input type="text"/>
Adresse	<input type="text"/>
Tél	<input type="text"/>
Nom d'utilisateur*	<input type="text"/>
Mot de passe*	<input type="text"/>
Confirmer mot de passe*	<input type="text"/>
<input type="button" value="Ajouter"/> <input type="button" value="Annuler"/>	

Figure: Interface ajouter un nouveau client

❖ **Interface contrôle des utilisateurs :**

A partir de cette interface, l'administrateur peut bloquer ou débloquer des clients.

Messagerie Interface Admin Deconnection

Contrôles des Utilisateurs

Voir: 10 Ligne	Recherche:		
Login	Nom	Prenom	
anis50	anis1	bellazreg	bloqué
riadh	Bellazreg	Riadh	debloqué

Voir 1 à 2 de 2 Ligne Debut Avant Suivant Fin

Figure: Interface contrôle des clients

❖ **Interface gestion messages contact**

Elle affiche une liste des messages contact dont l'administrateur a le choix entre répondre ou supprimer les contacts. Il y a deux types de contacts (clients et visiteurs).

User	Email	Sujet de messages	Messages	Date envoi			
ahmed	ahmed.bensalah1@hotmail.fr	contactez moi	besoin des informations	2012-05-05 03:03:42			
ahmed	ahmed.bensalah1@hotmail.fr	reclamation	pas recu la confirmation de l'inscription	2012-04-10 00:00:00			
anis50	bensalah1fr@yahoo.fr	securite	comment modifier mon mot de passe	2012-05-05 03:02:51			
mehdi	ahmed.bensalah1@hotmail.fr	education	demande de service pour education	2012-04-05 00:00:00			
najwa	ahmed.bensalah1@hotmail.fr	conseil	Bon courage	2012-05-01 21:23:59			
rawdha	ahmed.bensalah1@hotmail.fr	remerciement	je vous remercie	2012-05-05 03:06:24			

Figure: gestion de messages contact

REPONDRE AU CONTACT:

Email (obligatoire):
ahmed.bensalah1@hotmail.fr

Sujet:

Message:

Envoyer

Figure : Interface répondre

❖ Interface statistique de Google Analytics

Cette interface présente une interface sur google analytics qui permet de savoir le nombre de visiteurs de l'application.

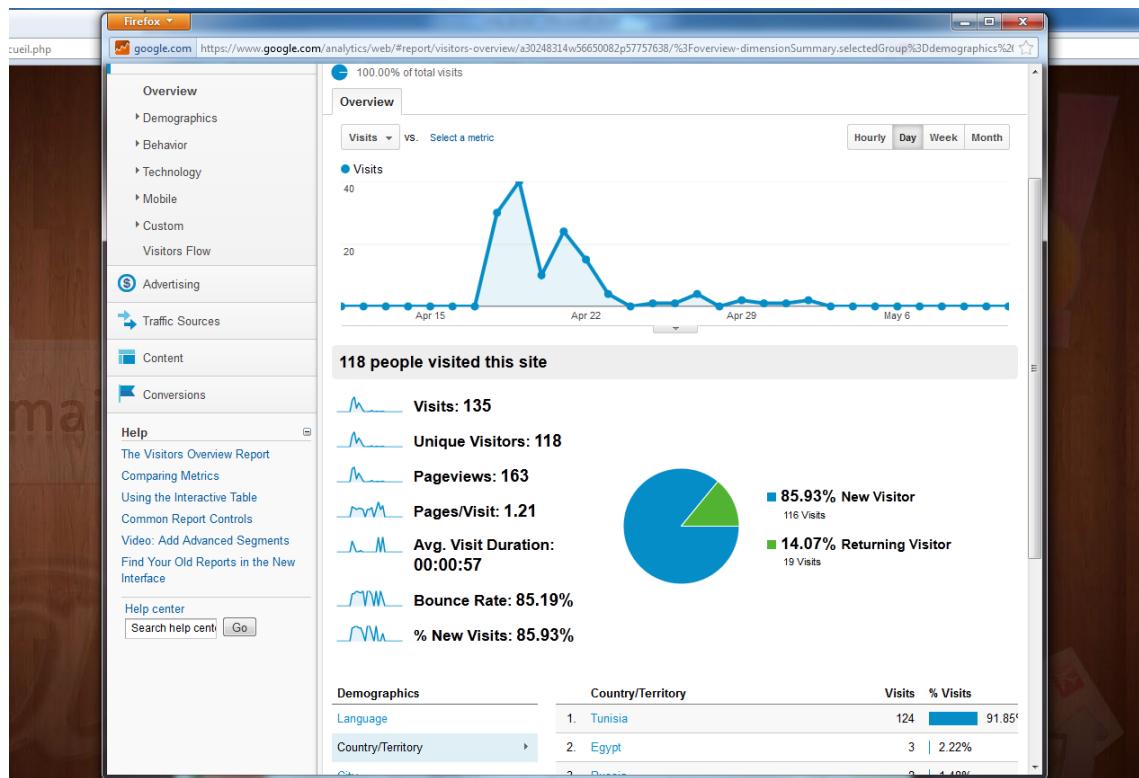


Figure: Interface statistique de Google Analytics

Conclusion générale

Pour conclure ce rapport, nous pouvons affirmer que ce stage se situe dans le prolongement des années de formations suivies. Il plonge l'étudiant dans un monde novateur où la pratique dépasse souvent la théorie. Une capacité d'autonomie est mise en avant pour progresser et faire face à de nombreux problèmes

Ce projet a pour objectif de présenter les différentes étapes de conception de cette application web ainsi que ses différentes composantes.

Cependant, ce projet constitue à la fois une étude théorique à travers la phase de conception et une étude pratique à travers la phase de réalisation de cette application.

La réalisation de ce projet, nous a permis de créer un système de gestion des données concernant notre application e-mailing, ainsi que la conception, qui constitue une phase importante dans la représentation des différentes fonctionnalités de l'application.

Cette application est extensible à être plus riche et on peut l'améliorer par plusieurs fonctionnalités qui facilite le contact entre les différents utilisateurs de l'application par exemple il est possible d'intégrer une boîte de réception ou une créer une boîte de messagerie instantanée.

En effet, ce travail nous a permis d'un côté d'enrichir nos connaissances théoriques à travers l'utilisation d'UML pour la conception orienté objet de systèmes d'informations, SQL comme un SGBD, et PHP comme un langage de programmation, d'une autre côté d'améliorer nos capacités de développement, afin d'en tirer profit dans le futur.

A titre général, ce stage a constitué une expérience professionnelle exceptionnelle. La vie universitaire ne permet que d'avoir un aperçu de son futur métier. On découvre rapidement que l'aspect pratique dépasse souvent les connaissances acquises et qu'il constitue une formation rapide.

Bibliographie

Sites Internet :

- [1]: <http://www.megacours.com/ccm/uml/umlintro.php>
- [2]: <http://www.aidenet.com/encyclopedie/uml/umlintro.htm>
- [3]: <http://mon.univmontp2.fr/claroline/backends/download.php?url=L3Byb2xlZ29tZW5lc191bWwucGRm&cidReset=true&cidReq=FMIN102>
- [4]: <http://laurent-audibert.developpez.com/cours-UML/html/cours-UML.html>
- [5]: http://fr.wikipedia.org/wiki/Diagramme_de_collaboration
- [6]: http://saoudyihab.voila.net/cours.uml/Diagramme_d_activite.pdf
- [7]: <http://dico.developpez.com/html/989-Conception-diagramme-de-deploiement.php>
- [8]: <http://fr.wikipedia.org/wiki/Hypertext Markup Language>
- [9]: http://www.3ie.fr/nouvelles_technologies/fiche/fiche_PHP.htm
- [10]: <http://fr.wikipedia.org/wiki/Photoshop>
- [11]: http://fr.wikipedia.org/wiki/Adobe_Dreamweaver
- [12]: <http://www.memoireonline.com/11/10/4012/Mise-en-place-d'un-système-d'information-pour-la-gestion-des-ressources-au-sein-de-la-DGI.html>
- [13]: <http://www.wampserver.com>
- [14]: <http://fr.wikipedia.org/wiki/PhpMyAdmin>
- [15]: <http://guides.ovh.com/introductionApache>
- [16]: <http://www.wampserver.com>
- [17]: <http://fadace.developpez.com/poweramc>
- [18]: http://fr.wikipedia.org/wiki/Rational_Rose

Les ouvrages :

De UML à SQL - de Christian Soutou

UML pour l'analyse d'un système d'information - de Chantal Morley