

Rapport de projet



Une application de messagerie électronique

Réalisé par :

- BOUDCHAR Nawal
- DIALLO Imrana

Table des figures

Figure 1 : le modèle incrémental	5
Figure 2 : Diagramme de cas d'utilisation	7
Figure 3 : Diagramme de séquence de l'authentification d'un utilisateur	8
Figure 4 : Diagramme de séquences de l'envoi des e-mails	9
Figure 5 : Les couches d'une architecture trois tiers	10
Figure 6 : L'application en trois tries	11
Figure 7 : Modèle Entité Association de l'application.....	12
Figure 8 : Diagramme de paquetage.....	13
Figure 9 : Diagramme de classe du package IHM	14
Figure 10 : Interface d'authentification.....	16
Figure 11 : Interface d'inscription	16

Table des matières

Table des figures	1
Table des matières	2
Introduction	3
Chapitre 1 : Contexte général	4
1. Le service de messagerie	4
2. Présentation du sujet	4
3. Cycle de développement	4
Chapitre 2 : Analyse et spécification des besoins	6
1. Analyse fonctionnelle	6
2. Analyse non fonctionnelle	6
2.1 L'ergonomie des interfaces	6
2.2 La sécurité et la fiabilité	6
3. Spécification détaillée	7
3.1 Le diagramme de cas d'utilisation	7
3.2 Diagrammes de séquences	8
3.2.1 Scénario nominal de l'authentification d'un utilisateur	8
3.2.2 Scénario d'envoi des e-mails	9
Chapitre 3 : Conception	11
1. Conception architecturale	11
1.1 Architecture trois-tiers	11
1.2 La correspondance entre le modèle trois-tiers et l'application	12
2. Conception détaillée	13
2.1 Conception de la base de données	13
2.2 Décomposition en paquetage	14
2.3 Diagrammes des classes	14
Chapitre 4 : Réalisation	15
1. Environnement logiciel	15
2. Interfaces de l'application	15
Conclusion	17

Introduction

Il ne fait désormais plus aucun doute que les technologies de l'information et de la communication représentent la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité en ce siècle.

Parmi les Technologies de l'information et de la communication(TIC) les plus utilisées on trouve la messagerie électronique qui caractérisée par sa facilité d'utilisation et son utilité perçue.

Le présent rapport s'articule autour de quatre chapitres. Dans le premier chapitre, nous présentons le cadre général de notre application ainsi la définition du service de messagerie. Le deuxième chapitre, s'intéresse à la partie analyse et spécification en se basant sur le langage UML (Unified Modeling Language). Nous consacrons le troisième chapitre à la partie conception qui comporte une illustration d'architecture de notre application et la conception détaillée de la base de données.

La réalisation c'est la dernière partie qui précise l'environnement du travail et présente les principales interfaces de l'application. Finalement nous clôturons le rapport par une conclusion générale qui présente le bilan de ce projet et les éventuelles perspectives.

Chapitre 1 : Cadre général

L'échange de courriers électroniques est certainement l'un des plus vieux et des plus utilisés de tous les services offerts sur Internet.

1. Le service de messagerie

Un service de messagerie, dans sa forme la plus basique, est un service permettant essentiellement l'échange de messages textuels entre les différents utilisateurs enregistrés (ayant une adresse électronique valide) et connectés à un réseau informatique, que ce soit en local ou sur internet. Mais, actuellement, les services de messagerie sont beaucoup plus riches et présentent beaucoup plus de fonctionnalités, à savoir l'intégration des pièces jointes (joindre un fichier quelconque au message envoyé), la gestion du courrier indésirable (les spams) et la manipulation des listes de diffusion (permettre l'envoi multiple).

Cet échange de messages peut s'effectuer en différé, c'est-à-dire il n'est pas nécessaire que le destinataire soit connecté au moment de l'envoi, son message sera enregistré sur un serveur et il pourra le consulter ultérieurement. On parle à ce moment de la messagerie électronique simple.

Par ailleurs, l'échange peut aussi se faire en temps réel et on parle à ce moment de la messagerie instantanée.

2. Présentation du sujet

Le but de ce projet est d'étudier et de développer un logiciel de messagerie qui offre les fonctions essentielles d'échange des e-mails (écrire, envoyer et recevoir).

Les utilisateurs de cette application doivent d'abord l'installer et le configurer avec le serveur.

3. Cycle de développement

Nous avons choisi le modèle Incrémental pour gérer le cycle de vie de notre projet parce qu'il permet de gérer les projets de développement de grands systèmes. Il découpe le système en domaines qui sont traités individuellement sur le modèle en cascade.

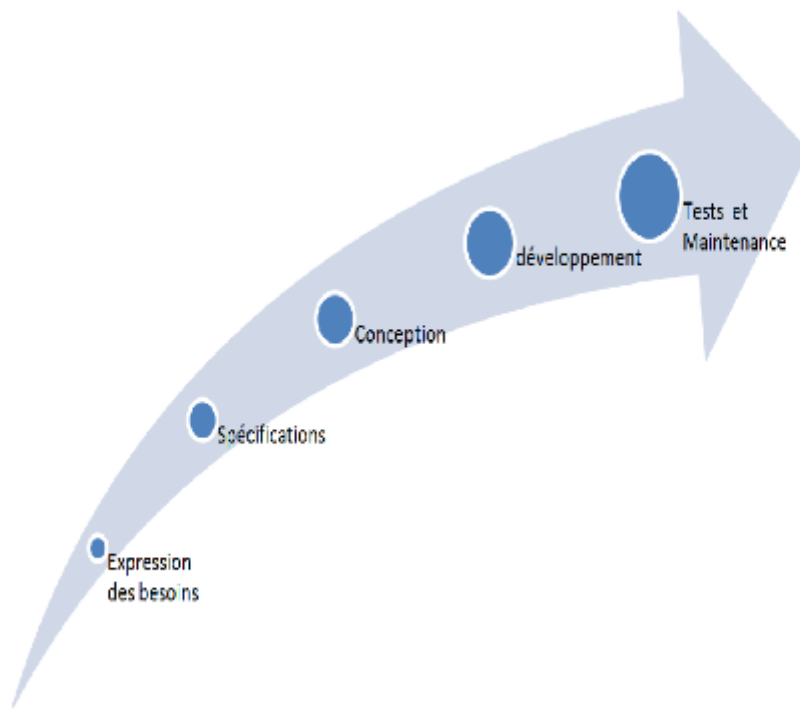


Figure 1 : le modèle incrémental

Chapitre 2 :

Analyse et spécification des besoins

Afin de garantir la réussite et l'efficacité de notre projet, il faut à ce stade du travail définir avec précision la bordure de la solution à développer. Ceci inclut l'énumération des différents services que notre système est supposé offrir aux différents utilisateurs.

1. Analyse Fonctionnelle

L'acteur principal de notre application est toute personne qui l'a installé, nous présentons dans ce qui suit les besoins de ce client.

Le client doit pouvoir jouir des fonctionnalités suivantes :

- S'inscrire : le client crée un compte propre à lui dans l'application.
- S'authentifier : se connecter.
- Récupérer le mot de passe oublié. Si l'internaute oublie son mot de passe, il sera automatiquement redirigé vers une page de récupération de ce dernier.
- Envoyer des messages
- Personnaliser le profil en modifiant l'image par exemple.
- Recevoir et visualiser des messages.
- Créer un carnet d'adresse.

2. Analyse non fonctionnelle

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, telle que les contraintes liées à l'environnement et à l'implémentation, les exigences en matière de performances, les dépendances de plate-forme, la facilité de maintenance, l'extensibilité et la fiabilité.

2.1. L'ergonomie des interfaces

Nous avons essayé de présenter une interface ergonomique englobant toutes les fonctionnalités offertes. En plus la manipulation de cette interface ne nécessite pas des connaissances poussées en informatique, elle est simple et claire afin de s'adapter aux connaissances informatiques de notre utilisateur.

2.2. La sécurité et la fiabilité

Notre application garanti l'intégrité des données à l'utilisateur c'est-à-dire qu'elle garde leur forme et leur contenu original. En outre, elle protège la confidentialité en assurant la validité de l'identité de l'utilisateur. Ceci se fait entre autres par le moyen d'un mot de passe assurant le contrôle d'accès. Notre système garanti aussi la fiabilité de la recherche des informations

3. Spécification détaillée

Afin de détailler les besoins précédemment spécifiés, une bonne réflexion autour du développement de notre application par un langage de modélisation comme l'UML (Unified Modeling Language) s'avère nécessaire. Nous utilisons alors dans la suite les diagrammes des cas d'utilisation et les diagrammes de séquences comme moyens de notre spécification.

3.1. Le diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation illustrent les relations entre les cas d'utilisation du système et l'ensemble des acteurs qui agissent sur le système. Ainsi ces diagrammes permettent de décrire le comportement du système de point de vue utilisateur.

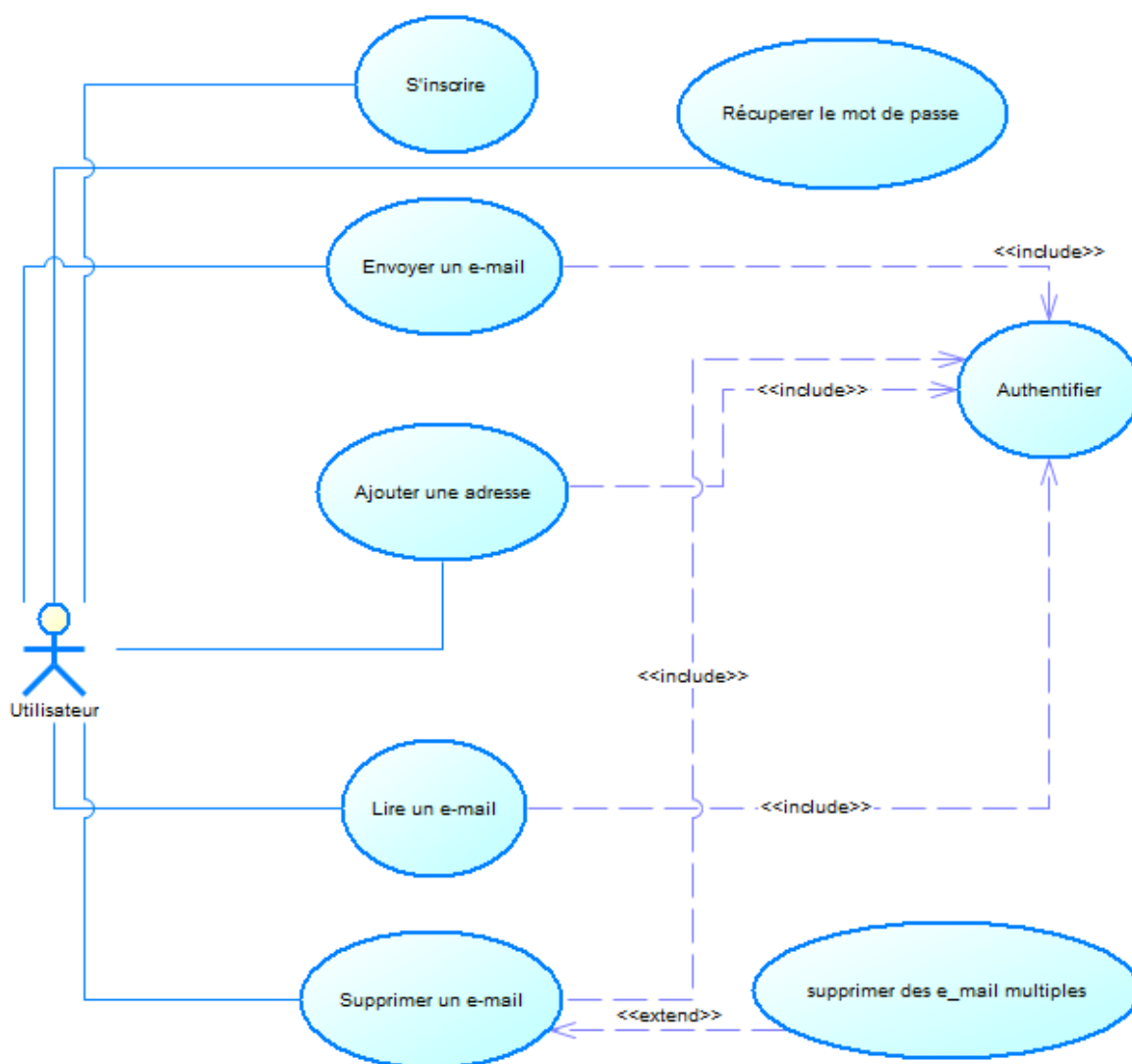


Figure 2 : Diagramme de cas d'utilisation

La fonctionnalité principale du système est l'échange des messages. Ainsi, un utilisateur peut écrire un message et l'envoyer directement à une destination choisie, dans ce cas

il garde aussi une copie (l'enregistrer). Il peut distinguer entre les messages lus et les messages non lus.

3.2. Diagrammes de séquences

Ces diagrammes sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Ces interactions sont ainsi montrées dans le cadre d'un scénario d'un diagramme des cas d'utilisation et ils ont pour but de décrire comment se déroule les actions entre les acteurs ou objets.

Ainsi, plusieurs diagrammes de séquences peuvent être représentés pour décrire le déroulement des différentes actions entre nos acteurs. D'autre côté, plusieurs diagrammes de séquences peuvent se rassembler, nous nous contenterons alors de présenter uniquement les diagrammes de séquences distincts.

3.2.1 Scénario nominal de l'authentification d'un utilisateur

L'accès au système se fait par le biais d'une adresse mail et d'un mot de passe. Ainsi, lors de l'appel de l'application, la page d'authentification se charge. L'utilisateur saisie ses paramètres personnels qui, après la validation, sont envoyés par l'objet <<Index-IHM>> vers l'objet <<Validation-Services>>.

Ce dernier a comme rôle de communiquer avec le serveur qui va vérifier l'existence de ces données. Par la suite, deux cas de figure se posent : L'échec ou la réussite de l'authentification. Dans le premier cas, un message d'erreur s'affiche pour informer l'utilisateur que soit l'adresse invalide soit le mot de passe incorrect. Concernant le cas de réussite l'utilisateur sera directement rediriger vers la page << Accueil >>.

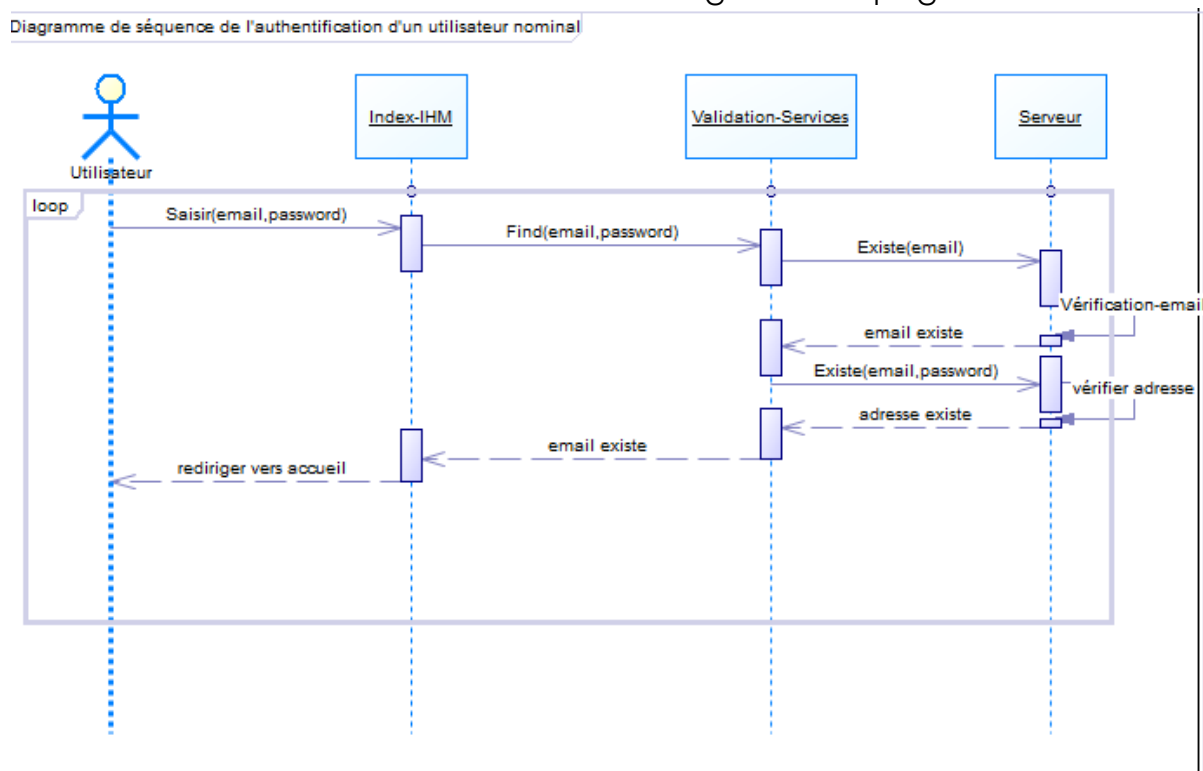


Figure 3 : Diagramme de séquence du scénario nominal l'authentification d'un utilisateur

2.2.2. Scénario nominal d'envoi des e-mails

Le diagramme de séquence ci-dessous illustre les interactions d'un utilisateur avec le système lors de l'envoi d'un message. L'utilisateur choisit le destinataire puis il écrit l'objet et le contenu du message. Lors de la confirmation d'envoi d'un message, l'objet << NouveauMessage-IHM >> envoie ces informations à <<Mail-Services>> qui va construire l'objet « Mail » et l'envoie au <<Serveur>>, Ce dernier prend en charge le stockage du message dans la boîte de réception de destinataire ainsi dans la boîte de messages envoyés de créateur.

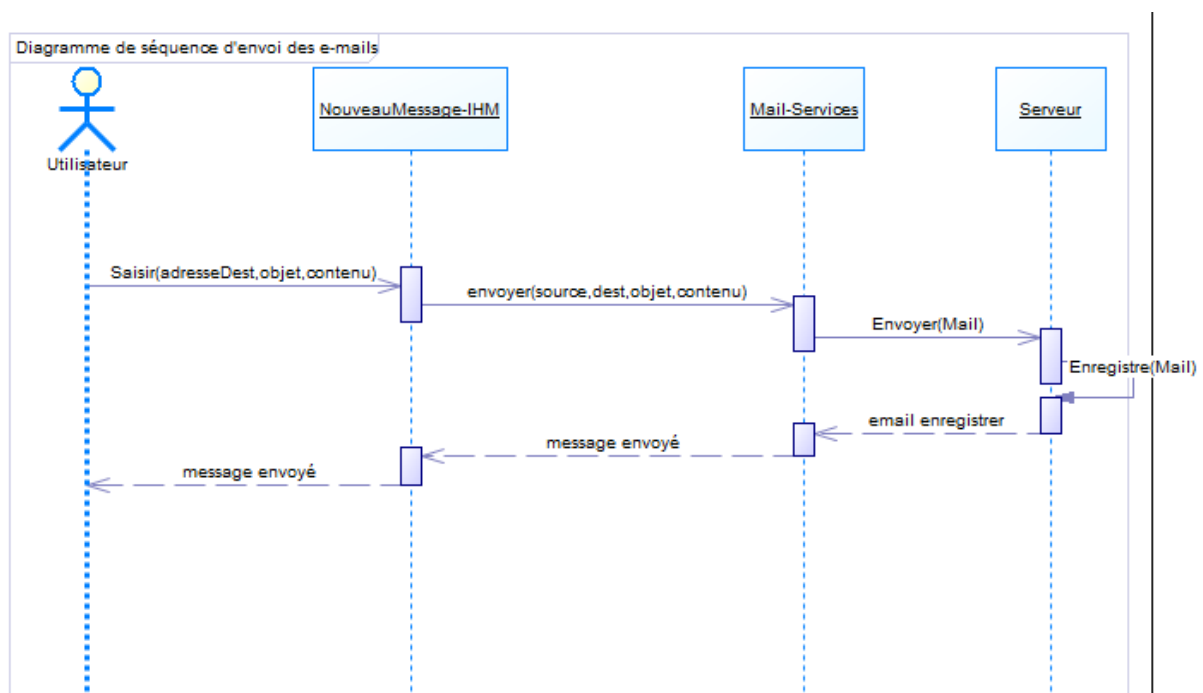


Figure 4 : Diagramme de séquences du scénario nominal d'envoi des e-mails

Chapitre 3 : Conception

Après avoir achevé la phase d'analyse et spécifications, nous entamons maintenant la phase de conception. Cette étape primordiale a pour but de détailler les tâches à entreprendre ainsi que de préparer le terrain pour l'étape de réalisation.

Pour ce faire, nous présentons une conception générale de l'application suivie d'une conception plus détaillée présentant le schéma de la base de données utilisée. Ensuite nous détaillons les différents modules de l'application aussi bien que les relations entre ces modules moyennant un diagramme de paquetage et un diagramme de classe. Enfin, nous exposons la cinétique de l'application.

1. Conception architecturale

Tout système d'informations nécessite la réalisation de trois groupes de fonctions : le stockage des données, la logique applicative et la présentation. Ces trois parties sont indépendantes l'une de l'autre : nous pouvons ainsi modifier la présentation sans modifier la logique applicative.

La conception de chaque partie doit également être indépendante. Toutefois la conception de la couche la plus basse est utilisée dans la couche supérieure.

Ainsi, la conception de la logique applicative se base sur le modèle de données, alors que la conception de la présentation dépend de la logique applicative.

1.1. Architecture trois-tiers

Le principe d'une architecture trois-tiers consiste à séparer la réalisation des trois parties vues précédemment (stockage des données, logique applicative et présentation).

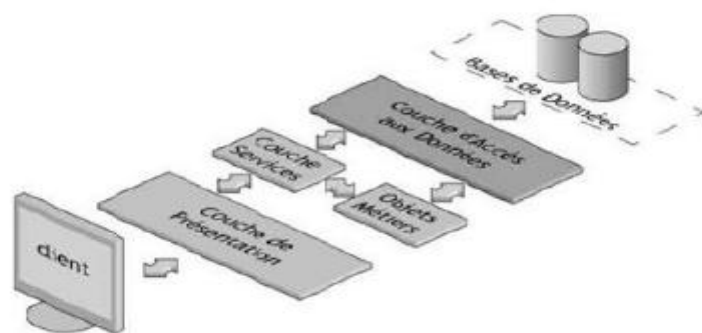


Figure 5 : Les couches d'une architecture trois tiers

- **Couche de présentation** : La présentation est la partie la plus immédiatement visible par l'utilisateur. Au niveau de cette couche se fait l'enregistrement, la récupération et la gestion des données persistantes dans une base de données.
- **Couche Services** : Cette couche réunit les traitements techniques, non fonctionnels qui sont pris en charge par le Framework de développement.
- **Objets métiers** : Ces objets font le travail essentiel lié au domaine de l'application. Ils nécessitent les traitements techniques, non fonctionnels de la couche service pour gérer la sécurité, le transactionnel, la concurrence.
- **La couche de persistance** : Elle est composée de la base de données. Le plus souvent on y ajoute une couche qui effectue la correspondance entre les objets et la base de données. Souvent cette couche sert aussi de cache pour les objets récupérés dans la base de données et améliore donc les performances.

1.2. La correspondance entre le modèle trois-tiers et l'application

La figure 6 met en évidence la conformité de notre application vis-à-vis du modèle trois-tiers.

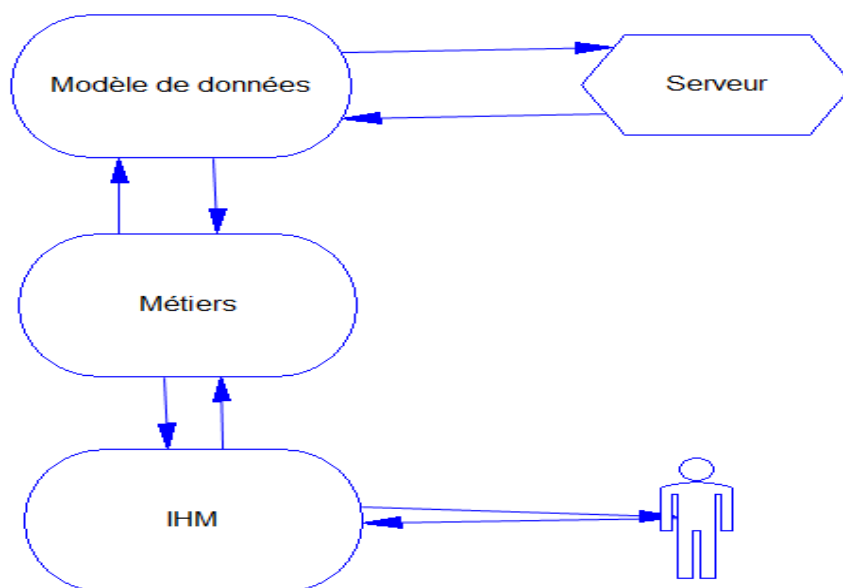


Figure 6 : L'application en trois-tiers

2. Conception détaillée

2.1. Conception de la base de données

Le schéma suivant explique la conception de la base en illustrant les relations entre les tables.

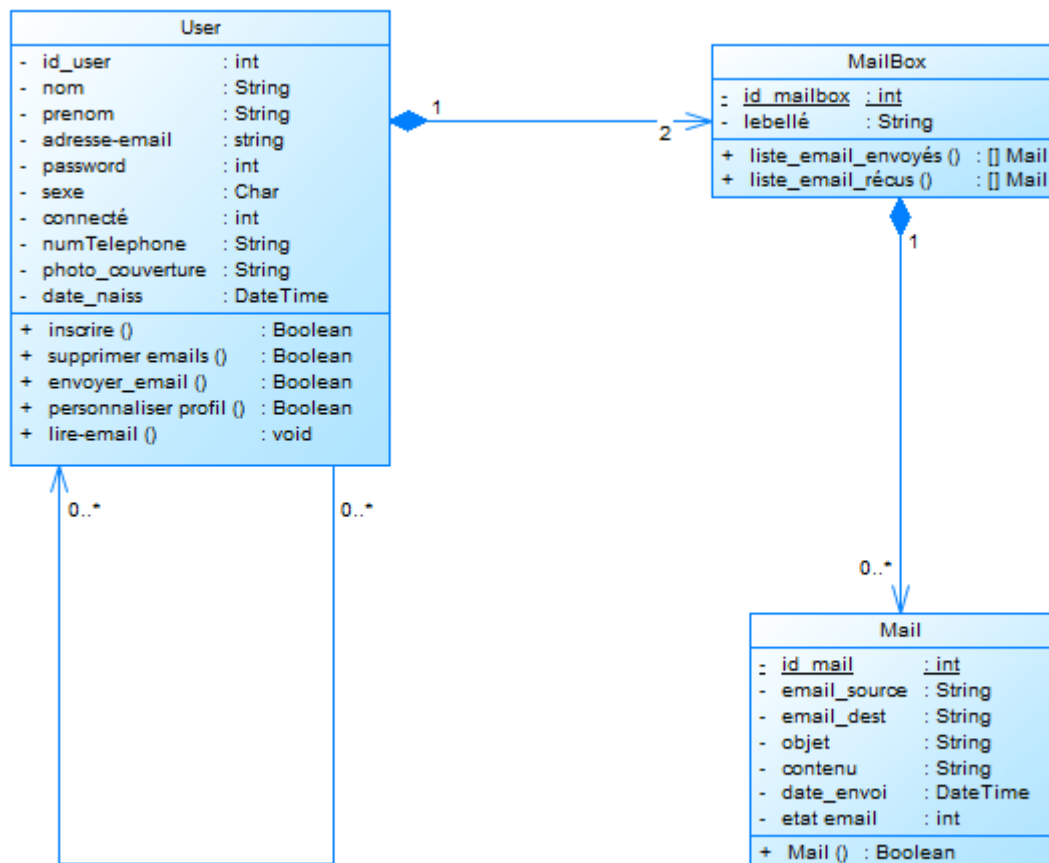


Figure 7 : Modèle Entité Association de l'application

➤ Présentation des tables :

- L'entité « **USER** » décrite par un id, le nom et le prénom de l'utilisateur ainsi qu'un mot de passe pour l'authentification contient également d'autres informations supplémentaires. L'utilisateur peut s'inscrire, envoyer des messages et profiter d'autres services.
- **Table Mailbox** : regroupe la liste des boîtes pour chaque « User » on a une boîte de réception et une boîte de messages envoyés
- **Table Mail** : sauvegarde les emails échangés par les utilisateurs.

2.2. Décomposition en paquetage

Pour passer à la conception, nous nous fondons sur les principes de l'approche orientée objet. À cet effet, nous passons d'une structuration fonctionnelle via les cas d'utilisation, à une structuration objet via les classes et les paquetages.

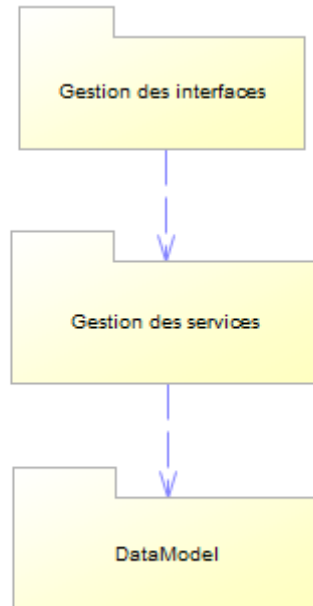


Figure 8 : Diagramme de paquetage

- **Package gestion des interfaces** : C'est le module qui contient toutes les interfaces qui vont communiquer avec l'utilisateur et qui vont gérer les actions de ce dernier tel que l'envoi des e-mails.
- **Package gestion des services** : représente le noyau de l'application, il regroupe les classes qui traitent les demande du premier package
- **Package DataModel** : contient la définition de la source de données.

2.3. Diagrammes des classes

La modélisation statique permet d'identifier, d'affiner et de compléter les différentes classes relatives aux paquetages de la section précédente. Elle consiste à rechercher les relations entre les classes et les compléter par leurs attributs spécifiques. Le diagramme de classes est le point central dans un développement orienté objet. Dans ce paragraphe, on va présenter à titre d'exemple le diagramme de classes représentant la couche présentation de notre application.

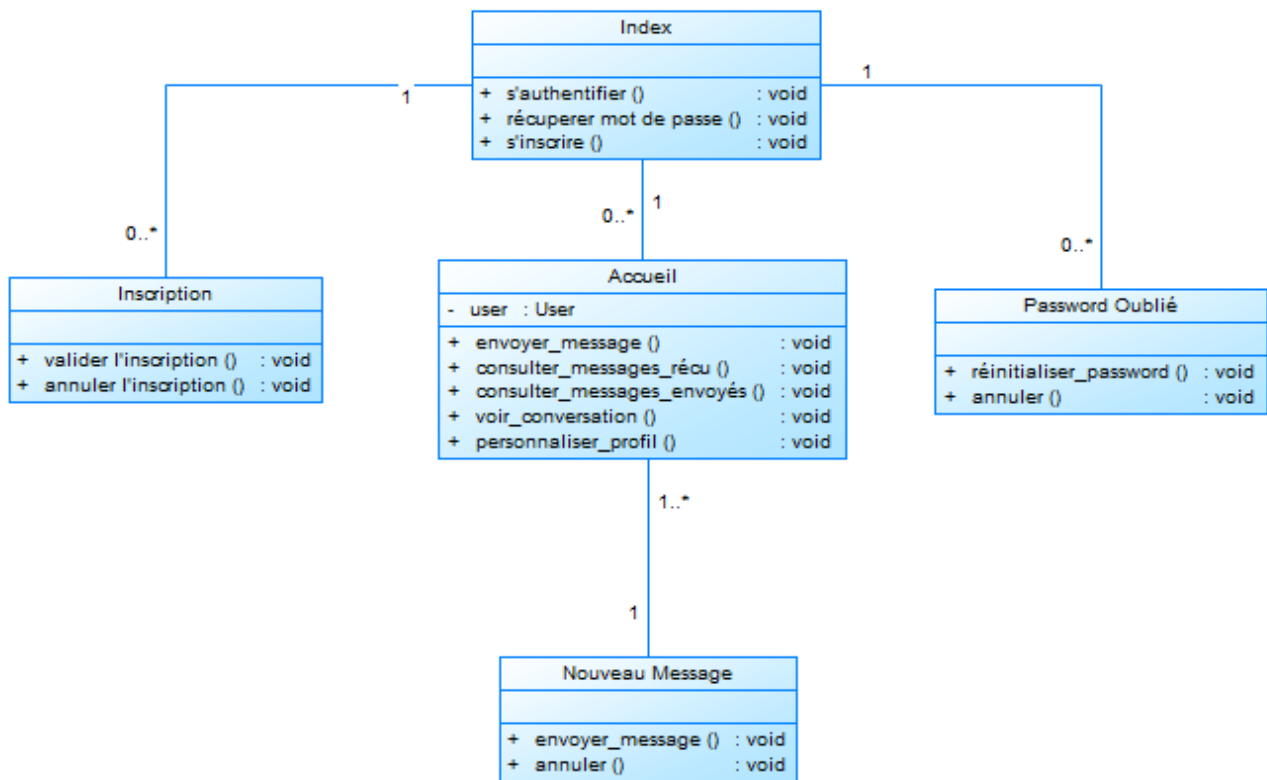


Figure 9 : Diagramme de classe du package IHM

Chapitre 4 : Réalisation

1. Environnement logiciel

- **Outils de développement** : Visual studio 2013
- **Conception et modélisation en UML 2** : PowerDesigner v16 est un outil de conception et de modélisation d'applications informatiques. Il permet aussi la modélisation des bases de données et la génération du code à partir des données de modélisation.
- **Data Base Server** : SQL serveur 2013
- **Désigne d'interface** : Photoshop CS6

2. Interfaces de l'application

L'interface graphique s'avère sans aucun doute la partie la plus cruciale dans une application.

La figure 10 illustre l'interface qui s'affiche au client désirant jouir des fonctionnalités de notre système. S'il dispose déjà d'un compte, il n'a qu'à fournir ses paramètres, à savoir mot de passe et adresse électronique, pour accéder à son profile. Dans le cas contraire, l'utilisateur est invité à s'inscrire.

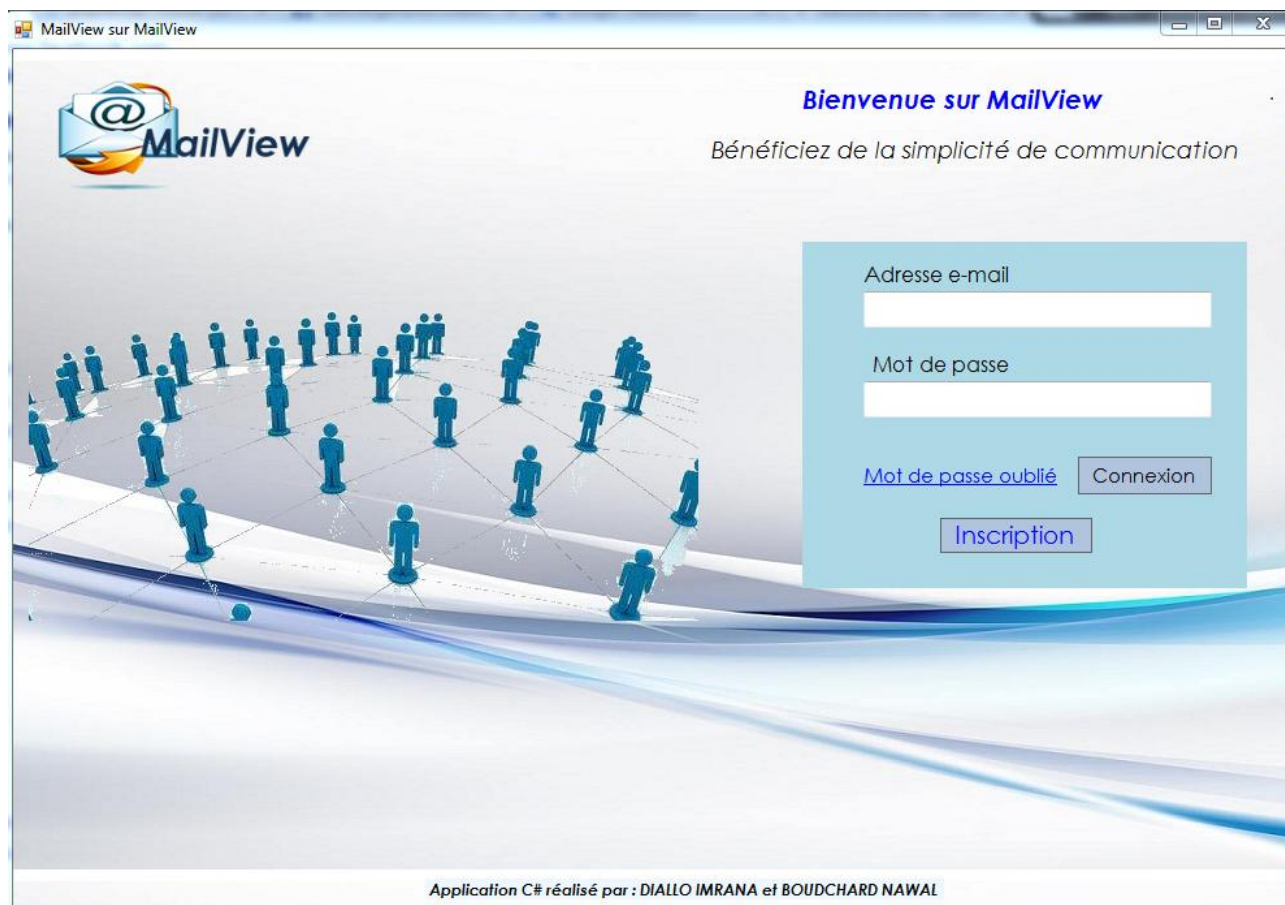


Figure 10 : Interface d'authentification

La figure suivante les champs du formulaire d'inscription :

 The image shows a web application window titled "Inscription". It contains a registration form with the following fields: "Nom :", "Prénom :", "Adresse électronique :", "N° de téléphone", "Mot de passe :", and "Confirmer le mot de passe :". Each field has a corresponding text input box. The "Adresse électronique" field contains the placeholder text "votreTexte@mailview.ma", and the "N° de téléphone" field contains the placeholder "XX-XX-XX-XX-XX". At the bottom of the form, there are two buttons: "Annuler" and "Confirmer".

Figure 11 : L'in

Conclusion

Ce projet était une bonne occasion pour nous de sortir du cadre théorique et appliquer les connaissances acquises en différentes matières, notamment l'utilisation du langage UML.

il nous a permis d'assimiler la technologie .NET par pratique, les linq to sql et le les fonctionnements d'une application client/server