République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Développement d'un annuaire téléphonique avec Java RMI

Réalisé par :

- **Mamchaoui Benaouda**
- **Baba Ahmed Adil**

Présenté le 10 Juin 2013 devant la commission d'examination composée de MM.

- Mr Benmammar B. *(Encadreur)*
- Mr Bendaoud F. *(Co-Encadreur)*
- Mr Chouiti S. *(Examinateur)*
- Mme Kazi A. *(Examinateur)*

Année universitaire : 2013-2014

Remerciement

A l'issue de ce travail, nous remercions, en premier lieu, le bon dieu de nous avoir donné la force et le courage de le mener à terme.

Nous tenons à remercier également toutes les personnes qui nous ont aidées de près et loin à la réalisation de ce projet parmi eux nous citons :

- Mr. Badr Benmammar , notre encadreur, qui nous a conseillés et guidés du début du projet à sa fin.
- Mr. F.Bendaoud, notre co-encadreur, qui nous a conseillés et guidés du début du projet à sa fin.
- Mme A. Kazi, notre examinateur, qui a accepté d'examiner notre projet.
- Mr. S. Mohamed. Chuity notre enseignant, examinateur, qui a accepté d'examiner notre projet.

DEDICACES

A peine nous venons de terminer la rédaction du mémoire de fin de cycle de licence, je voudrais très vite le dédier avec une immense joie, un grand honneur et un cœur chaleureux :

- *A mes très chers parents en signe de ma profonde et affectueuse reconnaissance pour leur amour sans mesure, tous les sacrifices, les soutiens, les tolérances et les encouragements qu'ils ont bien voulu consentir pour moi. Tous les mots restent faibles pour leur exprimer mes sentiments et qu'ils acceptent seulement ces lignes en guise de témoignage ;*

- *A Ma petite sœur;*

- *Enfin à tous mes ami(e)s et connaissances ;*

Baba Ahmed Adil

DEDICACES

A peine nous venons de terminer la rédaction du mémoire de fin de cycle de licence, je voudrais très vite le dédier avec une immense joie, un grand honneur et un cœur chaleureux :

- *A mes très chers parents en signe de ma profonde et affectueuse reconnaissance pour leur amour sans mesure, tous les sacrifices, les soutiens, les tolérances et les encouragements qu'ils ont bien voulu consentir pour moi. Tous les mots restent faibles pour leur exprimer mes sentiments et qu'ils acceptent seulement ces lignes en guise de témoignage ;*

- *A mes frères;*

- *Enfin à tous mes ami(e)s et connaissances ;*

Mamchaoui Benaouda

SOMMAIRE

Introduction générale

Liste des acronymes……..……07

Liste des figures ……...08

I. CHAPITRE 1 : Architecture et outils utilisés

I- 1 Introduction……....12

I- 2 Langage Java ……...12

I.2.1 Les points forts de java ……....12

I- 3 L'IDE Netbeans……..…..13

I- 4 Les middlewares……...…13

I- 5 Le protocole RMI…….....15

I- 6 Notion de table de hachage…….…18

I- 7 Client/serveur ……...20

I.7.1 Principe ….....20

I.7.2 Caractéristiques d'un processus serveur…....20

I.7.3 Caractéristiques d'un processus client…….....20

I.7.4 Avantages……21
I.7.5 Inconvénients…… 21
I.7.5 Type de client……..22
I.8 Conclusion……23

II. CHAPITRE 2 : Présentation de l'application

II- 1 Introduction ……25
II- 2 Description de l'application…… ..25
II- 3 La structure de la table de hachage…….....…25
II- 4 L'interface de démarrage…….....26
II- 5 L'interface principale…….....……27
II- 6 L'interface ajouter…….....29
II- 7 L'interface rechercher …….....31
II- 8 exemple sur la recherche par nom……33
II- 9 exemple sur la recherche par numéro……..34
II- 10 conclusion……35

Conclusion générale

Bibliographie

Liste des acronymes

PC : Personnel Computer

PFE : Projet de Fin d'Etude

JSP : Java Server Pages

JVM **:** Java Virtual Machine

OS: Operating System

IDE: *Integrated Development Environment*

CDDL: Common Development and Distribution License

XML : Extensible Markup Language

HTML: Hypertext Markup Language

RPC: Remote Procedure Call

RMI: Remote Method Invocation

CORBA: Common Object Request Broker Architecture

OMG: Object Management Group

API: Application *Programming Interface*

RRL: remote Reference Layer

OSI: Open Systems Interconnection

Liste des figures

Figure 1.1 portabilité du code en JAVA……12

Figure 1.2 Position de la couche intermédiaire middleware……..14

Figure 1. 3 Role de Middleware dans l'échange de données ……15

Figure 1. 4 Déploiement de RMI ……17

Figure 1. 5 Etapes d'un appel de méthode distante ……18

Figure 1. 6 Passage d'une requête client avec RMI……18

Figure 1. 7 Un annuaire représenté comme une table de hachage……..19

Figure 1. 8 Model demande/reponse……21

Figure 1. 9 Model client léger avec un serveur edubuntu ……22

Figure 1. 10 Model client lourd avec un serveur de base de données……23

Figure 1.11 Réseau Peer to Peer……23

Figure2.1 : structure de la table de hachage de l'annuaire……25

Figure2.2 : démarrage de l'annuaire……26

Figure2.3 : présentation du projet……27

Figure2.4 : la barre des menus……28

Figure2.5 : message d'information ……28

Figure2.6 : bouton ajouter……28

Fihure2.7 : bouton rechercher……28

Figure2.8:l'interface principale……29

Figure2.9 : interface d'ajout……30

Figure2.10 : message d'enregistrement……...30

Figure2.11 : message d'erreur d'un numéro déjà enregistré……...31

Figure2.12 : message d'erreur d'un champ vide…….31

Figure2.13 : interface de recherche……..32

Figure2.14 : la saisi d'un mode de recherche …….32

Figure2.15 : message d'erreur mode de recherche non sélectionné ……...33

Figure 2.16 : champ de saisi du mot clé …….33

Figure2.17 : tableau d'affichage des contacts…….33

Figure2.18 : exemple de recherche par nom……...34

Figure2.19 :exemple de recherche par numéro ……..34

Introduction générale :

Commençons par dire que la technologie est le seul élément qui peut mesurer et déterminer une société moderne. En effet l'homme essaye d'adopté de nouvelles techniques studieuses et révolutionnaires afin de pouvoir créer une ère plus moderne et confortable pour lui, c'est pourquoi, il faut toujours s'inspirer du passé pour trouver les meilleures idées adéquates à son nouvel environnement.

Notamment le monde informatique le confirme, puisque actuellement rien ne peut fonctionner sans pc et sans logiciel, ce support de connaissances occupe un rôle primordial au niveau de la communication et de la gestion des entreprises grâce à son volume qui est passé de la macro à la micro et son cout raisonnable, chaque individu peut le procurer.

On peut affirmer que cette révolution est la clé du développement.

- L'objectif de notre PFE est de faire la conception et la réalisation d'un annuaire téléphonique qui consiste à faciliter la manipulation d'un nombre conséquent de contacts afin de les enregistrer et les rechercher dans un répertoire téléphonique d'une façon simple et efficace.

Nous avons organisé ce mémoire de la façon suivante :

- Le premier chapitre présente l'ensemble d'outil et protocoles réseau utilisés dans la conception de notre application.
- Le deuxième chapitre présente l'implémentation de notre application dans laquelle nous illustrerons les différentes parties de l'application à savoir la table de hachage et les méthodes invoquées.

Chapitre I

Architecture et outils utilisés

I.1 introduction

Dans ce chapitre, nous allons présenter l'ensemble des outils nécessaire à la réalisation de notre PFE, nous allons présenter également l'architecture réseau utilisée ainsi que les différentes étapes de conception et de réalisation de notre application.

I.2 **Langage Java :**

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp). [1]

I.2.1 **Les points forts de java :**

- **La simplicité** : la syntaxe de java est une simplification de celle de C++.
- **Portabilité** : quand on compile un fichier java, il est transformé en bytecode exécutable sur toutes les plates-formes (JVM).

Comme indiquer dans la figure suivante :

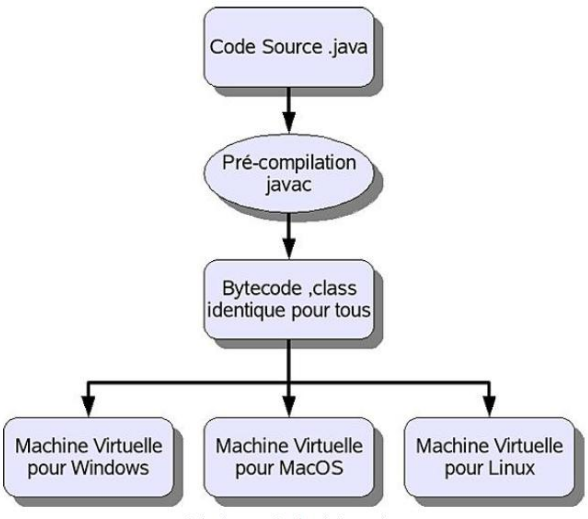

Figure 1.1: **portabilité du code en JAVA** [2]

- **Orienté objet :** Programmation puissante simplifiant entre autres le maintient de l'application. [3]

I.3 L'IDE Netbeans :

Présentation :

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Développent and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web). NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS. NetBeans est lui-même développé en Java, ce qui peut le rendre assez lent et gourmand en ressources mémoires. L'EDI NetBeans fournit pas mal d'outils pour supporter le développement d'application Java. [4]

I.4 Les middlewares :

La communication dans une application répartie se fait de deux manières :

a)- Communication bat niveau : Socket.

b)-Communication haut niveau : Middleware.

Un middleware est une couche intermédiaire (couche logiciel) qui s'intercale entre l'infrastructure de communication d'un réseau et les éléments de l'application distribuée. Ou autrement dit c'est une classe de logiciels permettant de faire la relation entre des couches de bas niveau d'un ordinateur (système d'exploitation) et des couches de haut niveau (applications). [5]

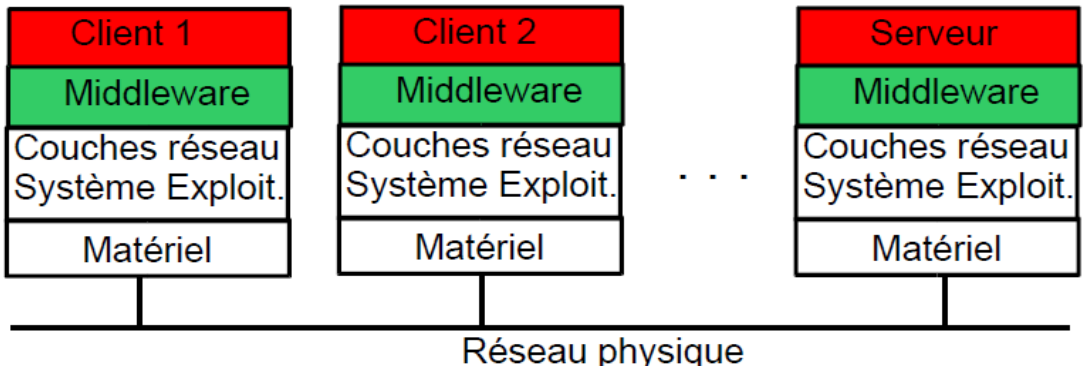

Figure 1.2: Position de la couche* intermédiaire *middleware [6]

En architecture informatique, un middleware (anglicisme) ou intergiciel est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques. Le réseau est mis en œuvre par l'utilisation d'une même technique d'échange d'informations dans toutes les applications impliquées à l'aide de composants logiciels.

Les composants logiciels du middleware assurent la communication entre les applications quels que soient les ordinateurs impliqués et quelles que soient les caractéristiques matérielles et logicielles des réseaux informatiques, des protocoles réseau, des systèmes d'exploitation impliqués.

Les techniques les plus courantes d'échange d'informations sont l'échange de messages, l'appel de procédures à distance et la manipulation d'objets à distance.

Les middlewares sont typiquement utilisés comme ciment pour relier des applications informatiques disparates des systèmes d'information des entreprises et des institutions. [7]

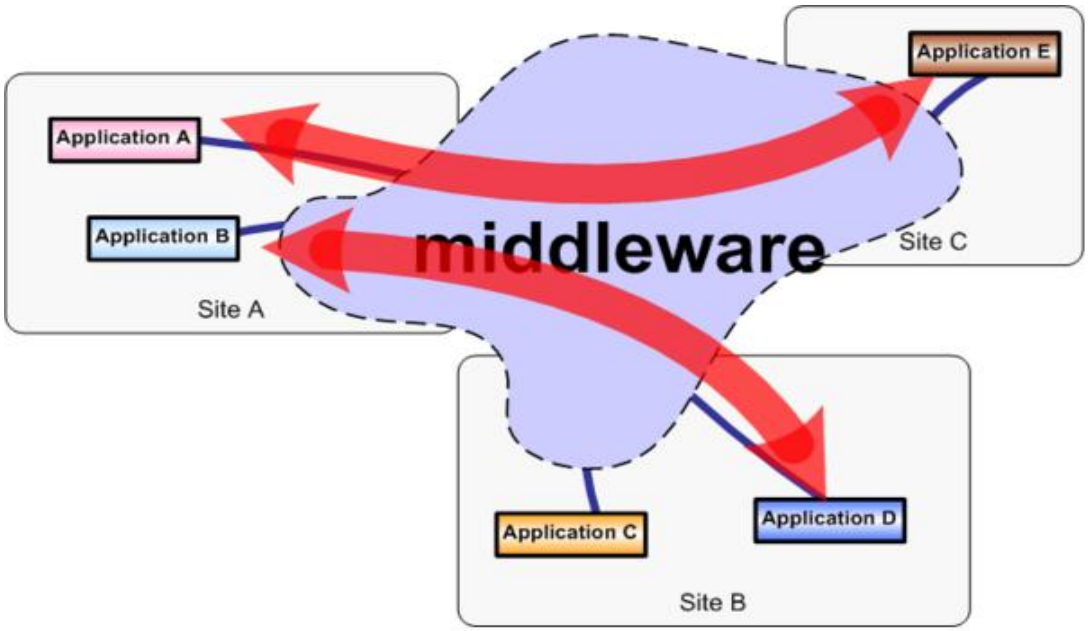

Figure 1.3: Rôle de Middleware dans l'échange de* données *entre les applications [8]

Il y a Plusieurs grandes familles de middleware et voilà quelques exemples : [9]

- ✓ RPC (Remote Procedure Call) : solution de Sun pour C/Unix.
- ✓ Java RMI (Remote Method Invocation) : solution native de Java.
- ✓ CORBA (Common Object Request Broker Architecture) : standard de l'OMG permettant l'interopérabilité quelque soit le langage ou le système

I.5 **Le protocole RMI**

L'utilité du protocole réseau RMI dans une application :

L'API RMI (Remote Method invocation) : est un mécanisme intégré dans java qui permet de mettre en œuvre facilement des objets distribués. RMI permet l'appel d'une méthode appartenant à un objet distant, c'est à dire gérée par une autre JVM (Java Virtual Machine) que la JVM locale : il devient ainsi possible à des applications dites clientes (s'exécutant localement) d'invoquer des méthodes sur des objets distants localisés dans une application appelée « serveur ». [10]

Principe et fonctionnement d'une application RMI :

Les applications RMI sont des applications bâties sur le modèle objet de java et dans lesquelles les objets sont répartis dans différents processus (en général sur différentes machines).

Une application RMI est composée d'un :

- Serveur : celui qui gère le service.
- Client : utilise le service.
- Service : le code lui-même ; répond aux clients.

- Dans un premier lieu la classe Serveur.java :

1- Crée une instance de l'objet demandé (le service).

2- Enregistré ce service dans l'annuaire rmiregistry.

Cela permettra aux clients de localiser les objets et d'obtenir des références vers ces objets.

La définition d'une interface qui contient les méthodes qui peuvent être appelées à distance ***(l'interface est partagée avec le client).***

- L'interface doit étendre **java.rmi.Remote.**
- Toutes les méthodes doivent lancer une **RemoteException.**
- Un objet qui implémente l'interface est un objet qui supporte l'accès distant à ses méthodes.

L'écriture des classes qui implémentent cette interface :

- Elles doivent implémenter l'interface.
- Elles doivent étendre UnicastRemoteObject.
- Il faut que les constructeurs lancent une RemoteException.

- Elles peuvent implémenter d'autres méthodes que celle dans l'interface mais celles-ci ne seront pas accessibles à distance.

- On deuxième position le Client.java :

Lorsqu'un client désire invoquer un objet à distance, il consulte le rmiregistry pour localiser l'objet, il fournit le nom de l'objet et reçoit en retour une référence vers cet objet. Avec la référence nouvellement obtenue, le client pourra invoquer les méthodes de cet objet. [11]

Voilà un schéma qui montre les différentes étapes d'un appel de méthode distante :

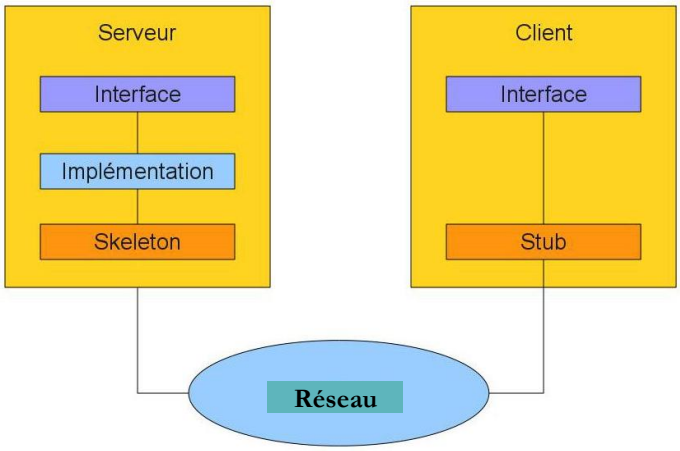

Figure 1.4: **Déploiement de RMI** [12]

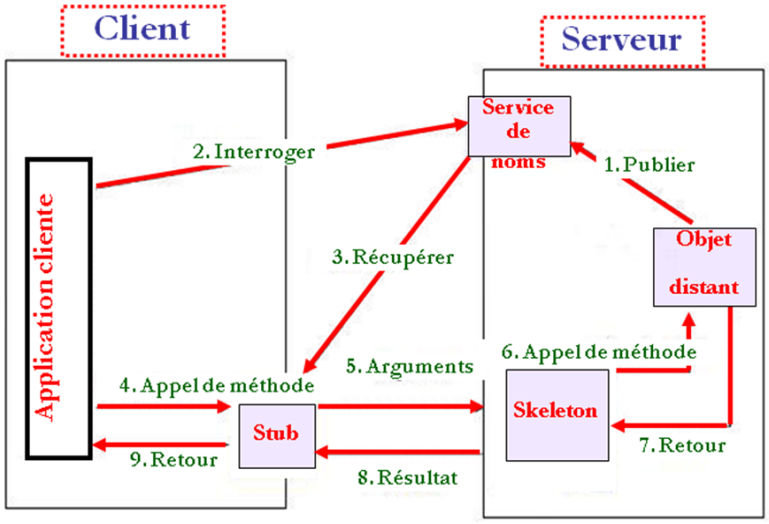

Figure 1. 5: **Etapes d'un appel de méthode distante** [13]

Les connexions et les transferts de données dans RMI sont effectués par Java sur TCP/IP grâce au protocole (JRMP, Java Remote Method Protocol) sur le port 1099 (port par défaut).

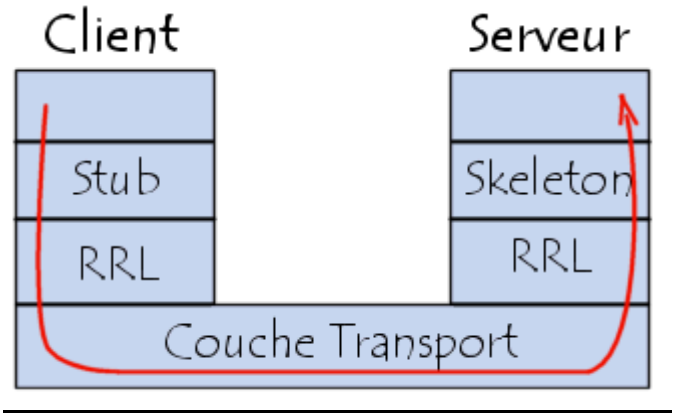

Figure 1.6: **Passage *d'une requête client avec RMI*** [14]

I.6 **Notion de table de hachage**

Définition

Une table de hachage est une structure de données qui associe des clés à des valeurs. Une telle structure doit au moins exposer deux fonctionnalités :

- Une méthode de type put (key, value), qui permet d'associer un objet à une clé ;
- Une méthode de type get (key), qui retourne la valeur qui a été associée à cette clé, ou null s'il n'y en a pas.
- Une méthode de type remove (key), qui supprime la clé de cette table, et la valeur qui lui est associée.

Il est bien sûr de la responsabilité de l'implémentation d'enregistrer les clés et les valeurs, et les associations qui existent entre elles.

En Java, une table de hachage entretient trois ensembles.

- Le premier est l'ensemble de ses clés. Il ne peut pas y avoir deux fois la même clé dans une table de hachage, cet ensemble ne peut donc pas posséder de doublon, il s'agit d'un Set.
- Le deuxième est l'ensemble de ses valeurs. À la différence des clés, une même valeur peut être associée à plusieurs clés différentes. Cet ensemble est donc une Collection.

- Le troisième est l'ensemble de ses entrées. Une entrée est un couple formé par une clé et la valeur qui lui a été associée. [15]

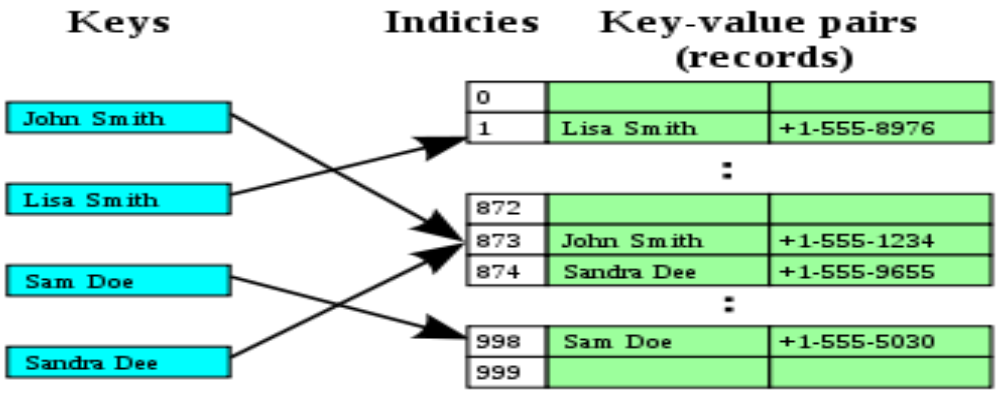

Figure 1. 7: **Un annuaire représenté comme une table de hachage.** [16]

I.7 **Client/serveur** :

I.7.1 Principe :

L'environnement **client-serveur** désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveur, attend les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.

En général, les serveurs sont des ordinateurs dédiés au logiciel serveur qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en termes de puissance de calcul, d'entrées-sorties et déconnexions. Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette), mais pas systématiquement. Un serveur peut répondre aux requêtes d'un grand nombre de clients.

Il existe une grande variété de logiciels serveurs et de logiciels clients en fonction des besoins à servir : un serveur web publie des pages web demandées par des navigateurs web ; un serveur de messagerie électronique envoie des mails à des clients de messagerie ; un serveur de fichiers permet de stocker et consulter des fichiers sur le réseau ; un serveur de données à communiquer des données stockées dans une base de données, etc.

I.7.2 Caractéristiques d'un processus serveur :

- Il attend une connexion entrante sur un ou plusieurs ports réseaux ;
- A la connexion d'un client sur le port en écoute, il ouvre un socket local au système d'exploitation
- Suite à la connexion, le processus serveur communique avec le client suivant le protocole prévu par la couche application du modèle OSI.

I.7.3 Caractéristiques d'un processus client :

- Il établit la connexion au serveur à destination d'un ou plusieurs ports réseaux ;
- Lorsque la connexion est acceptée par le serveur, il communique comme le prévoit la couche applicative du modèle OSI.

Le client et le serveur doivent bien sûr utiliser le même protocole de communication au niveau de la couche transport du modèle OSI. Un serveur est généralement capable de servir plusieurs clients simultanément. On parle souvent d'un service pour désigner la fonctionnalité offerte par un processus serveur. On définit aussi comme serveur, un ordinateur spécialisé ou une machine virtuelle ayant pour unique tâche l'exécution d'un ou plusieurs processus serveur.

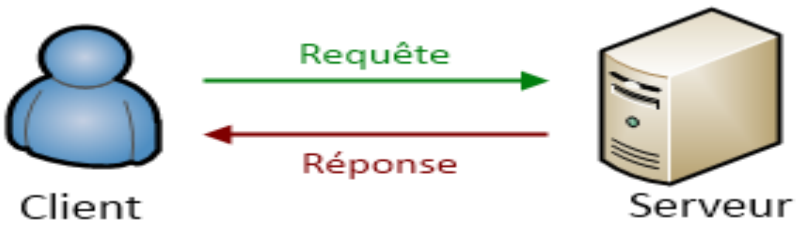

Figure 1.8: Model demande/reponse[17]

I.7.4 Avantages

- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.

- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.

I.7.5 Inconvénients

-Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge.

-Les coûts de mise en place et de maintenance peuvent être élevés.

-Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne

I.7.6 **Type de clients**

Client léger

Le poste client accède à une application située sur un ordinateur dit « serveur » via une interface et un navigateur Web. L'application fonctionne entièrement sur le serveur, le poste client reçoit la réponse « toute faite » à la demande (requête) qu'il a formulée.

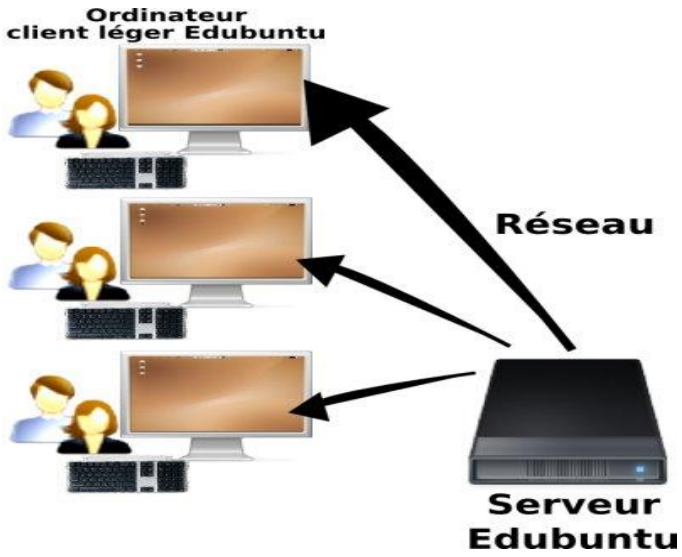

Figure 1.9: **Model client léger avec un serveur Edubuntu** [18]

Client lourd ***:***

le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un travail combiné entre l'ordinateur serveur et le poste client.

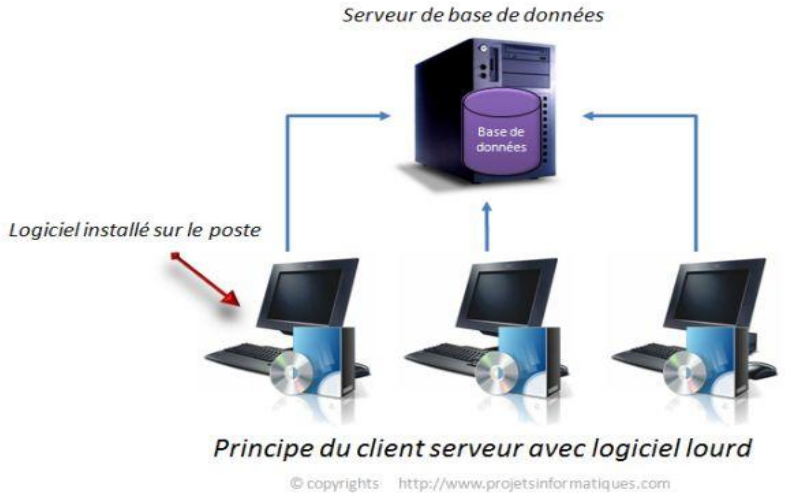

Figure 1.10: **Model client lourd avec un serveur de base de données** [19]

I.7.7 Architecture Peer to Peer

Les systèmes pair-à-pair permettent à plusieurs ordinateurs de communiquer via un réseau, en y partageant simplement des objets – des fichiers le plus souvent, mais également des flux multimédia continus (*streaming*), le calcul réparti, un service (comme la téléphonie avec Skype), etc.

Figure 1.11: Réseau **Peer to Peer** [20]

La particularité des architectures pair-à-pair réside dans le fait que les données puissent être Transférées directement entre deux postes connectés au réseau, sans transiter par un serveur Central. Il permet ainsi à tous les ordinateurs de jouer directement le rôle de client *et* serveur. On appelle souvent *nœud* les postes connectés par un protocole réseau pair-à-pair.

CONCLUSION

Ce chapitre a été dédié aux outils utilisés dans la partie conception, ces derniers nous on beaucoup aidé dans le développement de notre application. nous citons l'IDE netbeans, la table de hachage, le langage java et le middleware RMI. Nous les avons découvert et nous espérant les maitrisés pour concrétisé d'autres applications dans multiple domaines

Chapitre II

Présentation de l'application

II.1 Introduction

Ce chapitre est consacré à dévoiler tout le processus de développement de notre application annuaire téléphonique dans laquelle nous allons vous montré ces divers fonctionnalités et les principales étapes de réalisation du projet.

II.2 Description de l'application

C'est une étape primordiale au début de chaque démarche du développement. Son but est de veiller à mettre en valeur un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question :
Quelles sont les fonctions du système?
Notre système doit répondre aux exigences suivantes :

- L'insertion des clients dans le répertoire. cette dernière oblige l'utilisateur à inscrire 4 champs obligatoire (nom, prénom, mobile, N°fixe).

- Le système doit pouvoir récupérer des informations de chaque entité à partir de son matricule (mobile) pour mettre à jour la table de hachage de l'application.

- La recherche se fait en deux modes soit par mobile soit par nom, le client est dans l'obligation de choisir un mode sinon il recevra un message d'erreur.

II.3 La structure de la table de hachage

- Dans cette partie on va évoquer la structure de notre table de hachage, cet ordonnanceur d'entités comporte le mobile comme index et 4 autres champs nom, prénom, n°fixe et l'email.

mobile	nom	prénom	N°fixe	email
……...	…….	…….	…….	……..

Figure2.1 : Structure de la table de hachage de l'annuaire

Nous avons choisi le mobile comme étant index de notre table parce qu'il est unique à chaque client et parce qu'il abat le risque de conflit entre objets. Grace à ce schéma nous avons pu stocker nos contacts d'une manière sure et efficace.

II.4 L'interface de démarrage :

Au lancement de notre application on a opté pour une animation visuelle qui fait apparaître l'IHM suivante :

Figure2.2 : démarrage de l'annuaire

Cette image disparait en quelques secondes pour laisser sa place à une autre Frame qui va se stabiliser sur l'écran, cette dernière comporte le thème de notre application et un bouton (démarrer) qui fait office d'intermédiaire entre le démarrage et l'interface principale.

Figure2.3 : présentation du projet

En cliquant sur le bouton démarrer un son est distingué, ce battement annonce l'affichage de la nouvelle fenêtre qui est celle de la principale.

II.5 L'interface principale

Sur cette interface on peut apercevoir une barre de menu qui contient la partie **file** et la partie **?**

File : ce segment abrite 03 raccourcis :

- Ajouter (ctrl+A) : c'est un raccourci vers l'interface ajouter.
- Rechercher (ctrl+R) :c'est un raccourci vers l'interface rechercher.
- Quitter (ctrl+Q) :c'est un raccourci pour quitter l'application.

? : ce deuxième abrite qu'un seul raccourci (à propos Alt + f8).

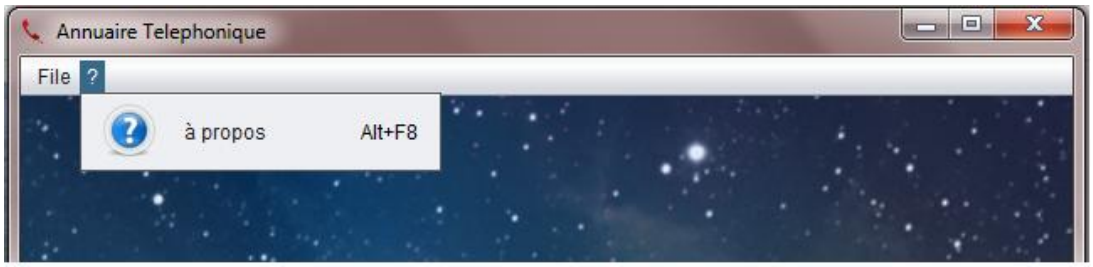

Figure2.4 : la barre des menus

Donc en cliquant sur le raccourci on va recevoir au retour un message nous indiquant le nom des concepteurs de cet annuaire téléphonique.

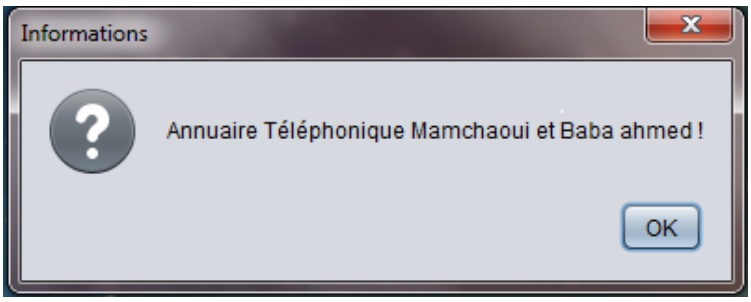

Figure2.5 : message d'information

On revient à notre interface principale, sur celle-ci on remarque la présence deux boutons :

Ajouter

Figure2.6 : bouton ajouter

: Ce bouton sert a basculé vers l'interface ajouter

Rechercher

Figure2.7 : bouton rechercher

: Ce bouton sert a basculé vers l'interface rechercher .

Voici une capture d'écran global sur l'interface principale.

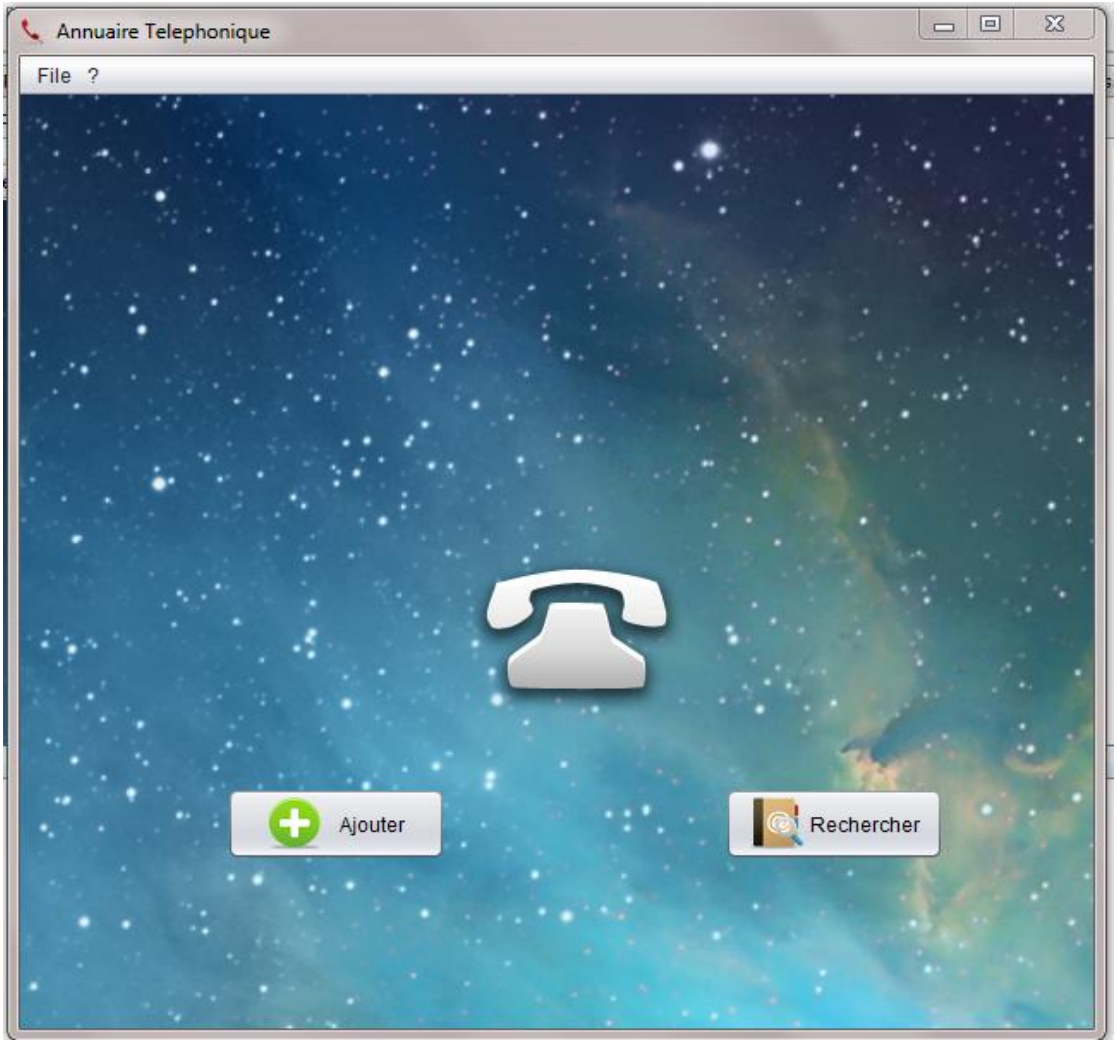

Figure2.8: l'interface principale

II.6 L'interface Ajouter

Sur cette interface on retrouve la où le client pourra enregistrer son contact en remplissant les champs de l'interface, on retrouve 4 champs obligatoire étoilé (nom, prénom, mobile, n°fixe) et un champ non obligatoire (email) non étoilé.

Voici une capture d'écran :

Figure2.9 : interface d'ajout

On retrouve aussi un bouton enregistrer qui sert a validé l'objet :

- Si l'objet est admis le client recevra un message lui indiquant qu'il a été enregistré dans la table de hachage.

Figure2.10 : message d'enregistrement

- Si le client enregistre 2 contacts avec le même numéro, il recevra un message d'erreur lui indiquant qu'il faut changer son numéro.

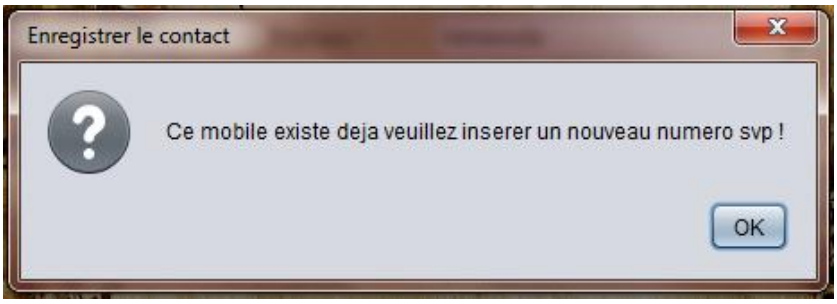

Figure2.11 : message d'erreur d'un numéro déjà enregistré

- Si l'un des champs obligatoire n'est pas rempli le client recevra un message d'erreur lui indiquant qu'il faut remplir un champ.

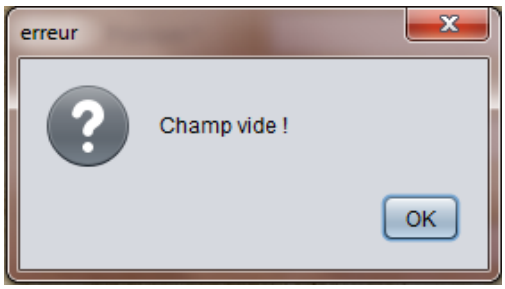

Figure2.12 : message d'erreur d'un champ vide

II.7 L'interface rechercher :

Voici un aperçu sur l'interface rechercher

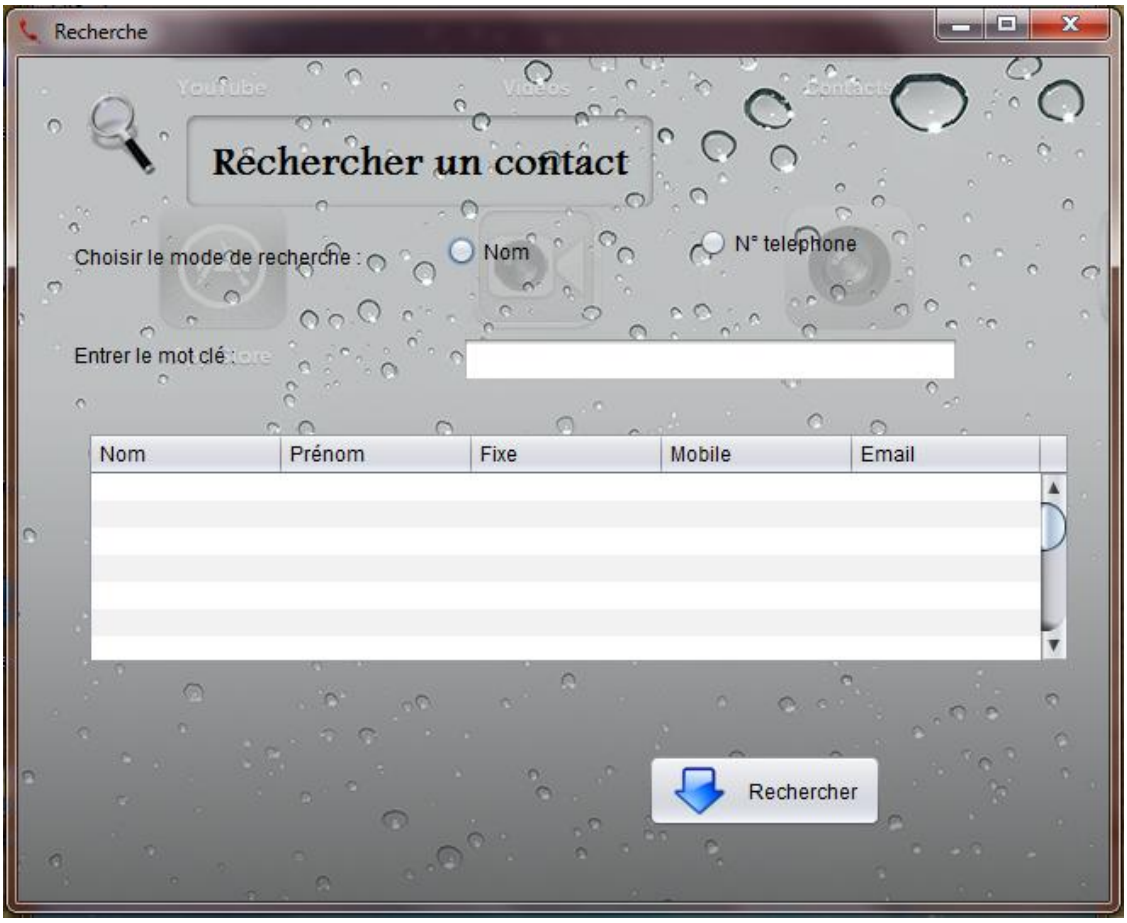

Figure2.13 : interface de recherche

Sur cette interface le client va suivre les étapes suivantes pour faire sa recherche :

L'utilisateur doit en premier lieu saisir un mode de recherche.

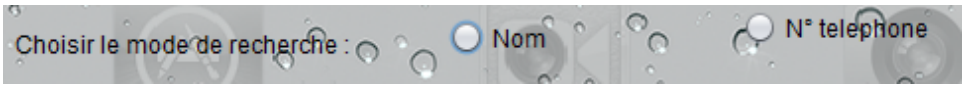

Figure2.14 : la saisi d'un mode de recherche

sinon il recevra un message d'erreur lui indiquant qu'il est obligé de choisir soit recherche par nom soit recherche par numéro.

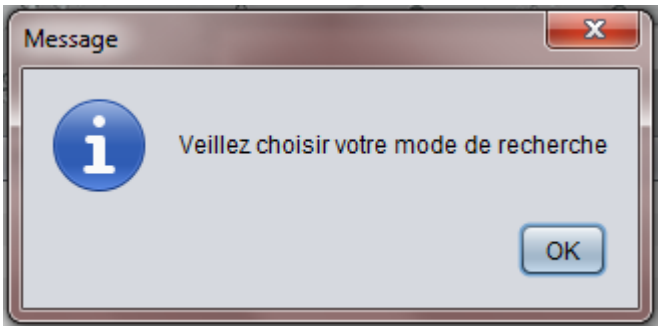

***Figure2.15* : message d'erreur mode de recherche non sélectionné**

Puis il doit entrer un mot clé dans le champ suivant :

Figure 2.16 : champ de saisi du mot clé

La recherche se fait de la manière suivante :

Soit par n° téléphone : le system va comparer le mot clé avec l'index (numéro) de la table de hachage s'il le trouve, il affichera l'objet dans le tableau suivant :

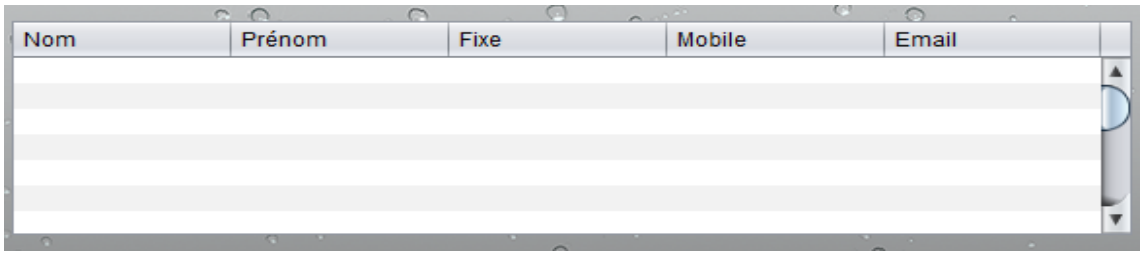

***Figure2.17* : tableau d'affichage des contacts**

Soit par nom : le système va comparer le mot clé avec les noms existant dans la table de hachage et retourner l'ensemble des noms invoqués.

II.8 Exemple sur la recherche par nom

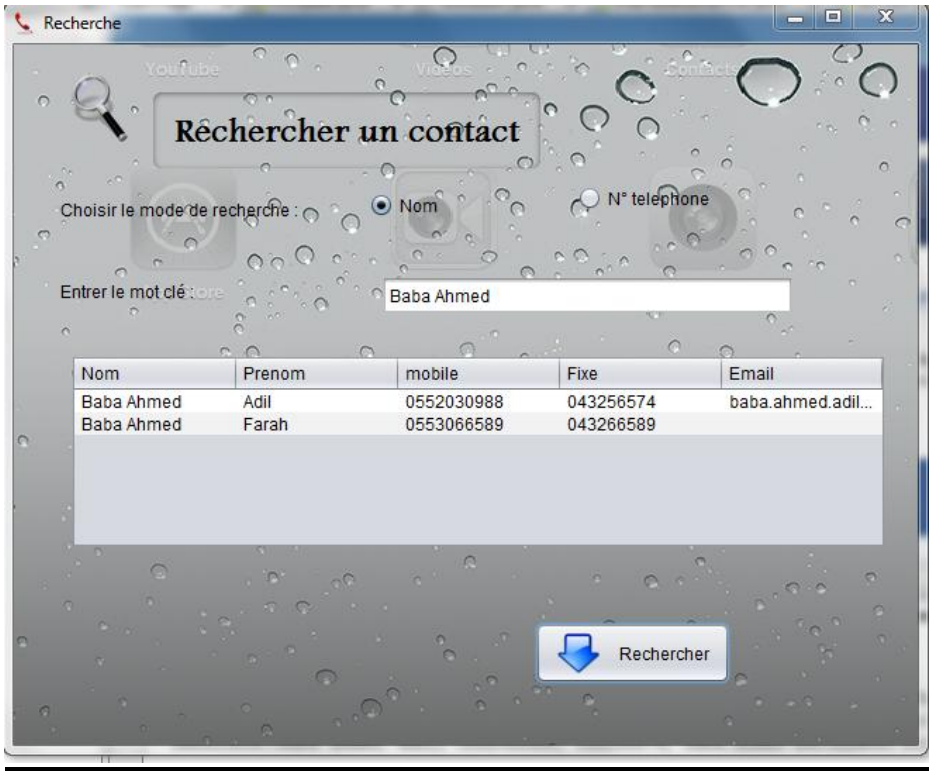

Figure2.18 **: exemple de recherche par nom**

II.9 Exemple sur la recherche par N°téléphone

Figure2.19 **: exemple de recherche par numéro**

II.10 Conclusion

Dans ce deuxième chapitre, nous avons présenté toutes les interfaces de l'application sous forme de capture d'écran en indiquant les tâches accomplies par chacune d'elle, de plus on a donné quelque exemple pour montrer les différents modes de recherches.

Conclusion générale

Au cours de ce mémoire, nous avons présenté les différentes étapes de conception et de réalisation de notre application (annuaire téléphonique en java RMI), Nous avons fait le choix de travailler avec le middleware RMI dans le but de découvrir un nouveau paradigme de programmation (orienté objet) et pour comprendre l'utilité d'un appel de méthode à distance tout en utilisant un model client/ serveur. Nous avons aussi utilisé l'IDE Netbeans pour exploiter sa flexibilité et son ergonomie et enfin on a exploité la table de hachage comme support de stockage.

Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation.

Cependant des perspectives d'améliorations de notre application restent envisageables.

Bibliographie

[1] http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-java-485/. Consulter le 11/05/2014

[2] Capture écran d'une vidéo Youtube.

[3] http://imac.epfl.ch/files/content/sites/imac/files/EnseignementTeaching/informatique/Aide2.1.pdf. Consulter le 11/05/2014

[4] http://www.bestcours.com/440-guide-pratique-edi-netbeans.html. Consulter le 11/05/2014

[5] http://ubimen.fr/dev/node/915. Consulter le 11/05/2014

[6] http://web.univ-pau.fr/~ecariou/cours/cs-umbb/cours-middleware.pdf. Consulter le 11/05/2014

[7] John Footen et Joey Faust, *The Service-Oriented Media Enterprise: SOA, BPM, and Web Services in Professional Media Systems*, Focal Press, 2008, (ISBN 9780240809779), chapitre 4, *definition of a middleware*.

[8] http://middleware.smile.fr/Concepts-des-moms-et-jms/Qu-est-ce-qu-un-middleware. Consulter le 11/05/2014.

[9] http://web.univ-pau.fr/~ecariou/cours/cs-umbb/cours-middleware.pdf. Consulter le 11/05/2014.

[10] Cours programmation réseau (RMI) n°9 - Université Paris Diderot. Consulter le 11/05/2014.

http://www.infres.enst.fr/~domas/TPRMI.html. Consulter le 11/05/2014.

[11] B Benmammar. Cours Introduction à JAVA RMI. M2 RSD. Page 37 Université de Tlemcen 2014.

[12] http://alain-defrance.developpez.com/articles/Java/J2SE/micro-rmi/. Consulter le 05/05/2014.

[13] B Benmammar. Cours Introduction à JAVA RMI. M2 RSD. Page 46. Université de Tlemcen 2014.

[14] http://www.commentcamarche.net/contents/1030-introduction-a-rmi-remote-method-invocation. Consulter le 11/05/2014.

[15] http://blog.paumard.org/cours/java-api/chap01-api-collection-map.html. Consulter le 11/05/2014.

[16] http://fr.wikipedia.org/wiki/Table_de_hachage .Consulter le 11/05/2014.

[17] http://lightcode.fr/livre-1/Les-reseaux/1-Introduction. Consulter le 11/05/2014.

[18] http://doc.ubuntu-fr.org/clientleger Consulter le 11/05/2014.

[19] http://300gp.ovh.net/~sitecoll/gpi3/site.php?rubrique=253. Consulter le 11/05/2014.

[20] http://doc.ubuntu-fr.org/p2p.

Résumé

Ce modeste travail sur le développement d'applications en java RMI nous a permis d'apprendre les différentes étapes de réalisation d'une application en se basant sur le model client/serveur. Ce travail décrit une étude de cas d'un développement d'un annuaire téléphonique. Il est découpé en deux chapitres : Le premier chapitre présente les différents outils misent en place pour le développement : le langage Java avec une interface de développement intégrée (IDE) Netbeans et une table de hachage utilisé comme support de stockage. Le chapitre 2 est consacré à la présentation de notre application en définissant les services proposés, et en présentant les différentes vues et interfaces.

MOTS CLES : JAVA, NETBEANS, middleware java RMI, table de hachage

Summary

This modest work on the development of applications in Java RMI has allowed us to learn the different stages of implementation of an application based on the client / server model. This report describes a case study of the development of a telephone directory. It is divided into two chapters: The first chapter presents the different tools put forward to the development with the Java language interface integrated development environment (IDE) Netbeans and a hash table used as a storage tool. Chapter 2 is devoted to the presentation of our application by defining the services offered, and presenting different views and interfaces.

KEYWORDS : : JAVA, NETBEANS, middleware java RMI, hashTable

ملخـــص

لقد سمح لنا هذا العمل المتواضع على تطوير تطبيقات جافا (إر .أم .إ)حيث سمح لنا بتعلم مختلف مراحل تنفيذ تطبيق بالإعتماد على نموذج العميل / الخادم. وتصف هذه الورقة دراسة حالة تطوير دليل الهاتف. وهي مقسمة الى فصلين : الفصل الأول يعرض الأدوات المختلفة التي طرحت للتطوير : لغة جاف مع واجهة التطوير المدمجة (بيئة التطوير المدمجة) نات بينس وجدول تجزئة يستخدم كوسيلة لعرض التطبيق لدينا من خلال تحديد الخدمات التي تقدمها، وتقديم وجهات 2للتخزين. ويخصص الفصل نظر مختلفة والواجهات.

الكلمات المفتاحية : جافا ,الوسيطة جافا (إر .أم .إ)، نات بينس , جدول تجزئة