Sigles et abréviations

Sigles et abréviations	Signification
2TUP	Two Track Unified Process
ATM	Automated Teller Machine
CU	Cas d'Utilisation
EDI	Environnement de Développement Intégré
OS	Operating System
PC	Personal computer
PDA	Personal Digital Assistant
SGBD	Système de Gestion de Base de Données
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
WSDL	Web Services Description Language
XML	eXtensible Markup Language

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Table des matières

Chapitre I : Présentation du projet 2
I. Présentation de la structure d'accueil 2
II. Présentation du Projet 2
1. Problématique 2
2. Objectifs 3
3. Les acteurs du projet 3
III. Le langage de modélisation 4
1. Présentation d'UML 4
2. Justification du choix d'UML 4
IV. Le processus de développement 2TUP 4
V. Le planning prévisionnel 5
Chapitre II : Etat de l'art de la biométrie 8
I. La biométrie 8
1. Système biométrique 8
2. Erreurs de système biométrique 9
3. Application de la biométrie 9
II. Les empreintes digitales 9
1. Forme générale de l'empreinte 9
2. Minuties ou motifs d'une empreinte digitale 10
3. Vue globale du mécanisme de reconnaissance des empreintes digitales 11
Chapitre III : Etude préliminaire 14
1. Les besoins fonctionnels 14
2. Les besoins opérationnels 14
3. Les choix techniques 15
Chapitre IV : Capture des besoins du système 18
I. Capture des besoins fonctionnels 18
1. Identification des fonctionnalités (Cas d'utilisations) 18
2. Le diagramme de cas d'utilisation 19
3. Description détaillée des cas d'utilisations 21
II. Capture des besoins techniques 34
1. Présentation du SDK MegaMatcher 34
2. L'architecture de déploiement 35
3. Les architectures réseaux 36

4. Coût de mise en œuvre 39
5. Coût de mise en production 40
Chapitre V : Spécifications conceptuelles 42
1. Généralité 42
2. Le diagramme de classe 42
3. Description textuelle du diagramme de classe 43
Chapitre VI : Réalisation 47
I. Configurations/Module pour PC 47
1. Configurations des outils d'implémentations 47
II. Configuration/Module pour Android 49
1. Configuration de l'IDE 49
2. Mise en place du web service 49
3. Procédure de déploiement/Module pour PC 50
III. Présentation de quelques écrans 51
1. Ecrans du Module pour PC 51
2. Ecrans du module pour Android 56

Liste des figures

Figure I.1 : Présentation 2TUP 5
Figure I.2 : Représentation graphique du planning 6
Figure II.1 : Les différentes familles d'empreinte digitale 10
Figure II.2 : Quelques minuties repérables sur une empreinte 11
Figure II.3 : Représentation d'une terminaison et d'une bifurcation 11
Figure II.4 : Traitement de l'empreinte 12
Figure II.5 : Processus d'enrôlement 12
Figure II.6 : Processus de vérification 12
Figure II.7 : Processus d'identification 13
Figure III.1 : Architecture fonctionnelle 15
Figure III.2 : Architecture de déploiement/Module Android 16
Figure III.3 : Modèle MVC 16
Figure IV.1 : Diagramme de cas d'utilisation 19
Figure IV.2 : Diagramme de cas d'utilisation/Module pour Android 20
Figure IV.3 : Diagramme de séquence: Enrôler empreinte 23
Figure IV.4 : Diagramme de séquence: Identifier empreinte digitale 26
Figure IV.5 : Diagramme de séquence: Vérifier empreinte digitale 29
Figure IV.6 : Diagramme de séquence: Rechercher doublons 31
Figure IV.7 : Diagramme de séquences : S'authentifier 33
Figure IV.8 : Architecture technique du SDK 34
Figure IV.9 : Architecture de déploiement/module pour PC 35
Figure IV.10 : architecture de déploiement/module pour Android 35
Figure IV.11 : Symboles /architecture réseau 36
Figure IV.12 : Architecture réseau/Module pour PC 37
Figure IV.13 : Architecture réseau/Module pour Android 38
Figure V.1 : Diagramme de classe 42
Figure VI.1 : Ecran de connexion/application sur PC 51
Figure VI.2 : Page d'accueil/application sur PC 52
Figure VI.3 : Ecran ajout personne 53
Figure VI.4 : Ecran ajout d'empreintes 54
Figure VI.5 : Ecran empreinte à identifier 55
Figure VI.6 : Ecran résultat identification 56
Figure VI.7 : Ecran de connexion/Application Android 57
Figure VI.8 : Ecran d'accueil/Application Android 58
Figure VI.9 : Ecran enrôlement/Application Android 58

Liste des tableaux

Tableau I.1 : Planning prévisionnel 6
Tableau III.1 : Liste des besoins fonctionnels 14
Tableau IV.1 : Tableau descriptif des différentes fonctionnalités 18
Tableau IV.2a : Description textuelle : Enrôler empreinte digitale 21
Tableau IV.2b : Description textuelle : Enrôler empreinte digitale 21
Tableau IV.3a : Description textuelle : Identifier empreinte digitale 21
Tableau IV.3b : Description textuelle : Identifier empreinte digitale 215
Tableau IV.4a : Description textuelle : Vérifier empreinte digitale 21
Tableau IV.4b : Description textuelle : Vérifier empreinte digitale 21
Tableau IV.5a : Description textuelle : rechercher doublons 30
Tableau IV.5b : Description textuelle : rechercher doublons 31
Tableau IV.6 : Description textuelle : S'authentifier 32
Tableau IV.7 : Liste des besoins logiciels/mise en œuvre 39
Tableau IV.8 : Liste des besoins matériels/mise en œuvre 39
Tableau IV.9 : Coût de mise en œuvre 40
Tableau IV.10a : Liste besoins logiciels/production 40
Tableau IV.10b : Liste besoins logiciels/production 40
Tableau IV.11 : Liste matériel minimal/production 41
Tableau IV.12 : Coût de production 41
Tableau V.1 : PersonneEnroler 43
Tableau V.2 : Utilisateurs 44
Tableau V.3 : Journal 44
Tableau V.4 : Session 45
Tableau V.5 : Profil 45
Tableau V.6 : Droits 45

Préambule

L'Université polytechnique de Bobo-Dioulasso (UPB), jadis Centre universitaire de Bobo-Dioulasso (CUPB), a été créée le 23 mai 1997 par décret n°97-54/PRES/PM/MESSRS. Située à une quinzaine de kilomètres à l'Ouest de Bobo-Dioulasso, elle est composée d'une école (L'Ecole Supérieure d'Informatique) de trois instituts (L'Institut de Développement Rurale, L'Institut Universitaire de Technologie, L'Institut des Sciences de la Santé) et de deux UFR (L'UFR Science et Technologies et L'UFR Sciences Juridiques et Politiques/Economiques et de Gestion) :

L'Ecole Supérieure d'Informatique (ESI) où nous avons suivi notre formation, a une organisation pédagogique qui s'articule autour des deux cycles suivants :

- la formation au Cycle des Ingénieurs de Travaux Informatiques (CITI) avec deux options dont « Analyse et Programmation » et « Réseaux et Maintenance Informatiques » ;

- la formation au Cycle des Ingénieurs de Conception en Informatique (CICI).

La formation en CICI a pour objectif de former des cadres supérieurs qui sont aptes à :

- gérer des centres informatiques ;
- gérer des projets informatiques ;
- participer efficacement à la conception, à la réalisation et à la maintenance d'applications informatiques ;
- assurer la formation des utilisateurs.

Durant leur cursus universitaire, les étudiants en CICI doivent effectuer un stage obligatoire au cours duquel ils réalisent leur projet de fin d'étude. L'objectif de ce stage est non seulement de permettre aux étudiants de mettre en exergue les connaissances acquises à l'école mais aussi de favoriser leur insertion professionnelle au sein des entreprises.

Introduction générale

La conception et la mise en place d'un système d'identification à l'échelle nationale revêt une importance capitale pour toute nation soucieuse de sa sécurité. Un système d'identification est un système permettant de renseigner des informations d'un groupe de personnes et d'effectuer facilement une recherche de ces personnes à travers des critères qui leur sont uniques.

Dans l'optique de moderniser et d'automatiser les processus d'accès aux informations de la police judiciaire, une base de données a été mise en place par des officiers de haut rang. Cette base de données existante qui est en cours d'expérimentation au niveau de la gendarmerie nationale permet essentiellement de répertorier des objets volés ou non, des objets perdus ou retrouvés, des personnes recherchées. L'objectif premier visé est de contribuer à la consolidation de la sécurité de l'Etat et de ses institutions. La multitude des critères de recherche rend l'identification très lente sans oublier l'impossibilité d'effectuer une identification rapide et fiable d'un individu en tout lieu. Comment rendre l'identification rapide, fiable et possible en tout lieu ?

Pour répondre à cette question et permettre l'extension et l'exploitation à l'échelle nationale du système, la gendarmerie nationale a mis en place un projet dénommé IRAPOL (**I**dentification, **Ra**pprochement, Analyse de **POL**ice).

Ce projet comporte un module biométrique permettant l'identification des personnes grâce aux empreintes digitales. Ce module devra fonctionner aussi bien sur des terminaux mobiles (smartphones, tablettes, etc..) que sur des micro-ordinateurs personnels.

C'est dans le souci d'offrir un outil fiable et performant d'aide à l'identification que nous avons été accueilli au sein de la société SITI-INFO pour un stage d'une durée de six mois (du 16 décembre 2013 au 16 juin 2014) sur le thème « ***conception et réalisation d'un système biométrique d'identification par empreinte digitale*** » qui comportera deux modules, notamment le module sur micro-ordinateurs personnels et le module sur terminaux mobiles (en particulier avec Android comme système d'exploitation).

Le présent rapport est subdivisé en six chapitres. Le premier chapitre sera consacré à la présentation du projet, le deuxième chapitre à l'état de l'art de la biométrie, le troisième chapitre à l'étude préliminaire, le quatrième chapitre à la capture des besoins du système, le cinquième chapitre à la spécification conceptuelle et le dernier chapitre à la réalisation.

Rapport-Gratuit.com

Chapitre I : Présentation du projet

Introduction

Dans ce chapitre nous faisons une brève présentation de notre structure d'accueil. Nous présentons le projet, son contexte et ses objectifs. Nous présenterons également le langage de modélisation ainsi que le processus de développement utilisé.

I. Présentation de la structure d'accueil

SITI-INFO, acronyme de Société Internationale de Technologie et d'Ingénierie Informatique, est une société de services et d'ingénierie informatique (SSII) au Burkina Faso. C'est une initiative de jeunes professionnels qui veulent partager leur expérience et leur savoir-faire pour offrir un plus au déploiement des technologies de l'information et de la communication au Burkina-Faso.

SITI-INFO a été créée en septembre 2009 avec pour objectifs immédiats:

- le déploiement d'une plateforme d'offre de services multimédia à forte valeur ajoutée ;
- la commercialisation de produits logiciels ;
- la réalisation sur mesure d'applications de gestions.

SITI-INFO a consolidé son expertise autour d'une plage de services diversifiés allant du développement spécifique d'applications au déploiement de technologies informatiques sûres et performantes.

Les services assurés par SITI-INFO sont : Centre d'appel, développement, intranet, décisionnel, réseau et système, sécurité et continuité de service.

II. Présentation du Projet

1. Problématique

L'application d'identification utilisée actuellement au sein de la gendarmerie nationale permet de recueillir les informations des personnes concernées et possède un module de recherche permettant d'identifier une personne déjà enregistrée dans la base de données. Pour effectuer ces recherches, plusieurs critères sont utilisés. Nous pouvons noter entre autres le nom, le prénom, la couleur des cheveux et des yeux, l'ethnie, le numéro de la CNIB, etc. Effectuer une identification devient alors une tache pénible compte tenu de la multitude d'informations requises pour identifier un individu.

Cette application ne fonctionnant que sur PC est très contraignante et cela pose un sérieux problème quant à l'identification des individus dans les zones rurales ou sur des lieux de scène (arrestation, vol, meurtre).

La lenteur et le manque de mobilité du système actuel posent un sérieux problème quant à l'identification à l'échelle nationale. Comment rendre l'identification rapide, en tout lieu et avec une grande fiabilité ?

Il convient donc de mettre en place un système biométrique basé sur les empreintes digitales qui est un système évolué et beaucoup utilisé de nos jours. Ce système devrait être fonctionnel aussi bien sur PC que sur les terminaux mobiles. Il devra permettre l'enrôlement à travers l'empreinte digitale et l'identification sera faite uniquement par empreinte digitale.

2. Objectifs

Le projet a pour objectifs de fournir :

- un outil efficace et fiable d'identification ;
- une base de données centralisée des données biométriques ;
- un système mobile.

3. Les acteurs du projet

a. *L'équipe de pilotage*

L'équipe de pilotage est composé de :

- M. KIENDREBEOGO Alain, Directeur général de SITI-INFO ;
- M. BAZIE Brahima, consultant à SITI-INFO.

b. *L'équipe de projet*

L'équipe de projet est constitué principalement de :

- M. WANGRE Mathias, stagiaire à SITI-INFO.

c. *L'équipe des utilisateurs*

L'équipe des utilisateurs sera constituée principalement des :

- agents de la police ;
- agents de la gendarmerie.

III. Le langage de modélisation

1. Présentation d'UML

UML (de l'anglais, Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, à spécifier, à concevoir des solutions et à communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objets.

2. Justification du choix d'UML

Les avantages présentés par UML, nous permettent de faciliter la modélisation des différents aspects de notre projet. En effet UML présente l'avantage d'être le standard de la modélisation objet le plus utilisé de nos jours. Sa notation graphique permet d'exprimer visuellement des solutions objets facilitant ainsi la comparaison et l'évaluation de celles-ci. C'est un langage formel et normalisé doté d'un gain de précision et d'un gage de stabilité. Il est aussi un support de communication performant car il cadre l'analyse tout en facilitant la compréhension des représentations abstraites complexes. En outre, UML sert à formaliser tous les documents techniques d'un projet et permet d'affiner les détails de l'analyse au fur et à mesure de l'avancée du projet. Il est possible d'utiliser le même atelier de génie logiciel depuis l'expression des besoins jusqu'à la génération de tout ou d'une partie du code. Enfin, il est indépendant des langages de programmation et des processus de développement.

IV. Le processus de développement 2TUP

2TUP signifie « 2 Track Unified Process» .C'est un processus qui répond aux caractéristiques du **P**rocessus **U**nifié. Le processus 2TUP apporte une réponse aux contraintes de changement continuel imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track» signifie littéralement que le processus suit deux chemins. Il s'agit des « chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.

Les caractéristiques de 2TUP sont :

- construit sur UML ;
- itératif et incrémental ;
- conduit par les cas d'utilisation ;

- piloté par les risques.

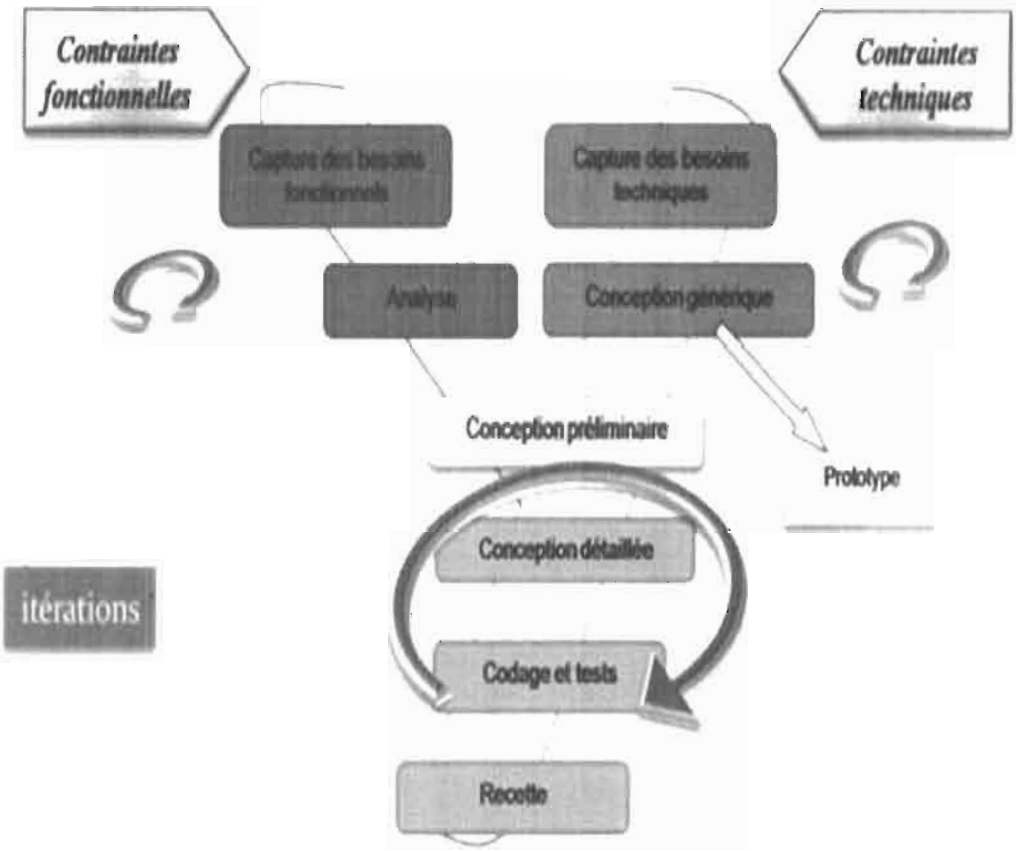

Figure I.1 : Présentation 2TUP [1]

V. Le planning prévisionnel

Il s'agit du planning prévisionnel suivi pendant ce stage. Nous allons faire une présentation textuelle sous forme de tableau puis sous forme graphique.

Le tableau I.1 présente le planning du stage, il présente les différentes tâches à exécuter, leur description et ainsi que la durée d'exécution. La figure I. 2 nous donne la représentation graphique du planning.

Tableau I.1 : Planning prévisionnel

Phase	Tâches	Description des taches	Durée	précédent
Accueil et connaissance du thème	T1	Connaissance de la structure d'accueil, du thème et adoption du planning	Du 16 au 23/12/13	
Etude préliminaire	T2	Définitions des besoins et fixations des objectifs	Du 24/12/13 au 7/01/13	
Capture des besoins	T3	Capture des besoins fonctionnels et techniques	Du 08/01/14 au 22/01/14	T1
Conception	T4	Conception de la base de données	Du 23/01 au 03/02/14	T2
Réalisation	T5	Configuration et implémentation des différents modules	Du 04/02/14 au 31/05/14	T3
Documentation	T6	Documentation des différents modules développés	Du 04/02/14 au 31/05/14	T4
Recettes unitaires	T7	S'assurer formellement que le produit est conforme aux spécifications	Du 02/06/14 au 16/06/14	T3

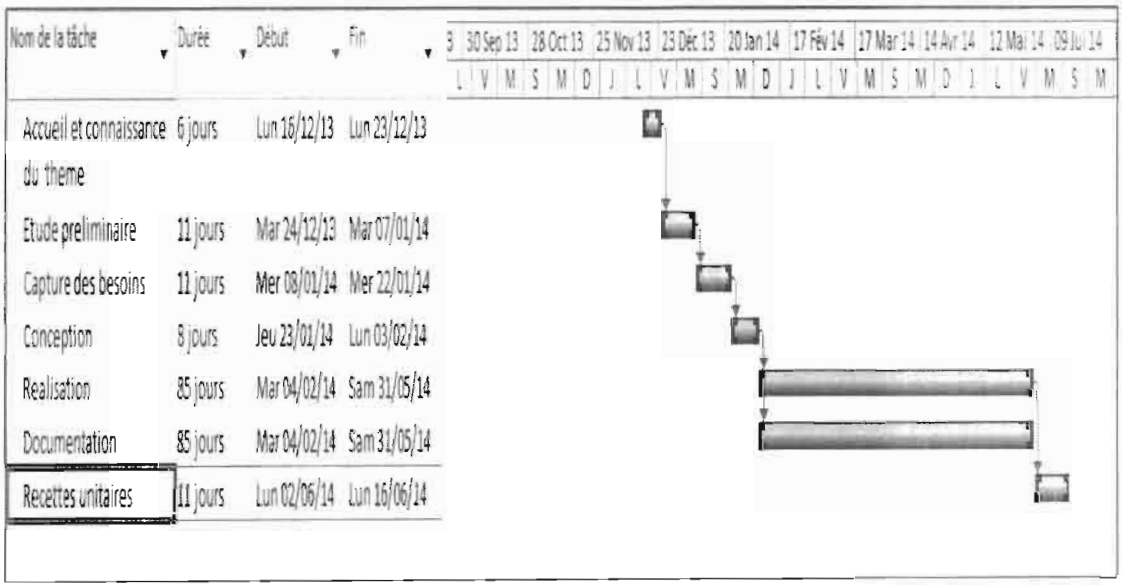

Figure I.2 : Représentation graphique du planning

Conclusion

Dans ce chapitre il a été question pour nous de présenter le projet globalement à travers la structure d'accueil, les acteurs ainsi que les objectifs du projet. Nous avons donné un aperçu sur le processus de développement 2TUP et le langage de modélisation UML. Il a été question également de présenter le planning prévisionnel suivi pendant le stage. Le chapitre suivant va nous permettre de mieux appréhender les notions relatives à la biométrie.

Chapitre II : Etat de l'art de la biométrie

Introduction

La biométrie est un ensemble de technologies (appelées les technologies biométriques) qui exploitent des caractéristiques humaines physiques ou comportementales telles que l'empreinte digitale, la signature, l'iris, la voix, le visage, ou la démarche pour différencier des personnes. À la différence des mots de passe ou des PIN (numéros d'identification personnelle) qui sont facilement oubliés ou exposés à l'utilisation frauduleuse, ou des clefs ou des cartes magnétiques qui doivent être portées par l'individu et sont faciles à être volées, copiées ou perdues, ces caractéristiques biométriques sont uniques à l'individu et il y a peu de possibilité que d'autres individus puissent remplacer ces caractéristiques . Les technologies biométriques sont considérées de nos jours comme étant les plus puissantes en termes de sécurité. En plus les mesures biométriques sont confortables parce qu'elles n'ont pas besoin d'être portées séparément. De telles caractéristiques peuvent être bien employées pour obtenir l'identification/authentification pour accéder à des systèmes tels que les ATM (guichet automatique), les ordinateurs et même pour l'accès aux immeubles. La biométrie se prouve également comme outil puissant d'identification/vérification sur les scènes de crime dans le secteur juridique.

I. La biométrie

1. Système biométrique

Un système biométrique est essentiellement un système de reconnaissance de formes qui fonctionne en acquérant des données biométriques à partir d'un individu, extrayant un ensemble de caractéristiques à partir des données acquises, et comparant ces caractéristiques contre la signature dans la base de données. Selon le contexte d'application, un système biométrique peut fonctionner en mode vérification ou/et en mode identification:

- Vérification: le système valide l'identité d'une personne en comparant ses données biométriques capturées à des données spécifiques déjà enregistrées dans la base de données.
- Identification: le système identifie un individu en comparant sa signature avec les signatures (*templates*) de tous les utilisateurs dans la base de données.

2. Erreurs de système biométrique

Un système biométrique fait principalement deux types d'erreurs:

- confusion des caractéristiques biométriques de deux personnes différentes pour être de la même personne,
- confusion des caractéristiques biométriques de la même personne pour être de deux personnes différentes.

Ces deux types d'erreurs se nomment souvent en tant que faute d'acceptation et faute de rejet, respectivement.

3. Application de la biométrie

La biométrie est appliquée dans plusieurs domaines et son application peut être divisée en trois groupes principaux :

- Applications commerciales: telles que l'accès au réseau informatique, la sécurité de données électroniques, le commerce électronique, l'accès d'Internet, l'ATM, la carte de crédit, le contrôle d'accès physique, le téléphone portable, le PDA, la gestion des registres médicaux, etc…
- Applications du secteur publique: telles que la carte nationale d'identification, le permis de conduire, la sécurité sociale, le contrôle de passeport, etc...
- Applications juridiques: telles que l'identification de cadavre, la recherche criminelle, l'identification de terroriste, etc...

II. Les empreintes digitales

1. Forme générale de l'empreinte

L'étude d'une empreinte digitale commence par l'observation de sa forme générale. Le but est de classifier les empreintes en trois grandes familles représentées par la figure II.1. Nous avons principalement les familles suivantes:

- **empreinte en boucle** vers la gauche.
- **empreinte en verticille** formant une sorte de tourbillon.
- **empreinte en arc** : les lignes sont disposées les unes au-dessus des autres en formant une sorte de A.

Empreinte en boucle **Empreinte en verticille** **Empreinte en arc**

Figure II.1 : Les différentes familles d'empreinte digitale [2]

2. Minuties ou motifs d'une empreinte digitale

On différencie les empreintes entre elles à l'aide de « points singuliers » :

- Points singuliers globaux :
  - noyau ou centre: lieu de convergences des stries ;
  - delta: lieu de divergences des stries.
- Points singuliers locaux (appelés aussi minuties) : points d'irrégularité se trouvant sur les points capillaires. Quelques minuties repérables sur une empreinte sont représentées à la figure II.2. On peut relever jusqu'à seize types de minuties mais dans les algorithmes on en retient que quatre :
  - terminaison à droite ou à gauche (minutie située en fin de strie) ;
  - bifurcation à droite ou à gauche (intersection de deux stries) ;
  - île : assimilée à deux terminaisons ;
  - lac: assimilée à deux bifurcations.

Les bifurcations et les terminaisons sont les deux types de minuties les plus utilisés car ils sont facilement détectables, mais surtout parce qu'ils sont très aisément représentables par le modèle de coordonnées, où chaque minutie est représentée par les coordonnées (x,y) de son emplacement et l'angle θ entre l'axe horizontal et la tangente à la strie (voir figure II.3). Les informations de chaque minutie d'une empreinte sont alors reprises dans un vecteur de la forme (type de minutie, x, y, θ), ce qui est une manière pratique de sauvegarder les informations d'une empreinte digitale sur un support informatique.

Figure II.2: Quelques minuties repérables sur une empreinte

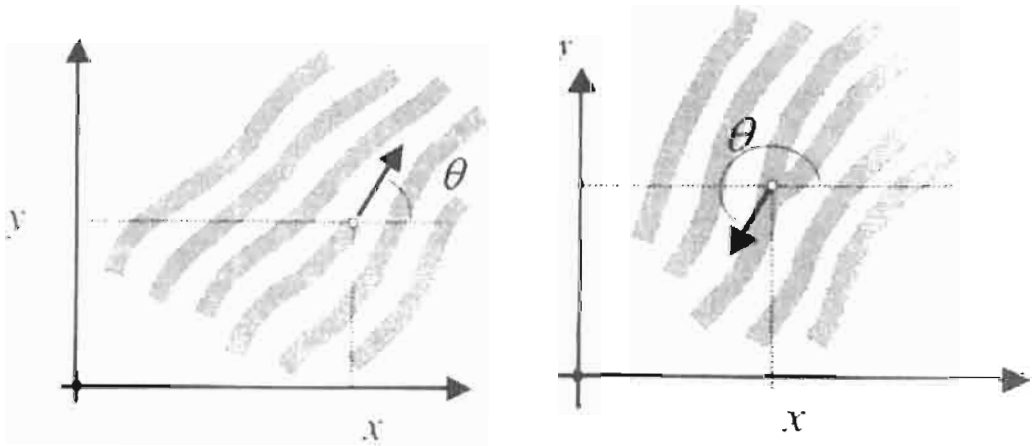

Figure II.3 : Représentation d'une terminaison et d'une bifurcation

3. Vue globale du mécanisme de reconnaissance des empreintes digitales

Le principe de la reconnaissance des empreintes digitales consiste à comparer une empreinte fournie au système, à une ou plusieurs autres empreintes (*les templates*) dont le système dispose préalablement dans sa base de données biométrique. Le système biométrique renvoie un résultat positif au cas où l'empreinte fournie correspond à l'un des *templates*, et un résultat négatif dans le cas contraire. La figure II.5 illustre l'enregistrement préalable de *templates*. Lors de l'enregistrement, l'image scannée de l'empreinte est recueillie par le système, puis un contrôle de qualité de l'image est effectué. En effet, une empreinte sérieusement endommagée (par une brûlure grave par exemple) est intraitable par le système. Ensuite, une extraction de traits caractéristiques (les minuties) est effectuée pour donner lieu au *template* final que le système sauvegarde dans la base de données. La méthode généralement utilisée pour détecter les minuties consiste à mettre l'image de l'empreinte en noir et blanc, c'est la *binarisation* de

l'image, et à donner une même taille aux lignes de l'empreinte, c'est la squelettisation. Une fois que l'on dispose de l'image binaire squelettisée (c'est-à-dire image dont les traits sont rendus fins), les minuties sont mieux visibles ; on procède alors à leur détection. La figure II.4 illustre les étapes de traitement de l'empreinte.

(a) Image d'origine. (b) Image binarisée. (c) Image squelettisée, minuties détectées (points en couleur).

Figure II.4 : Traitement de l'empreinte [3]

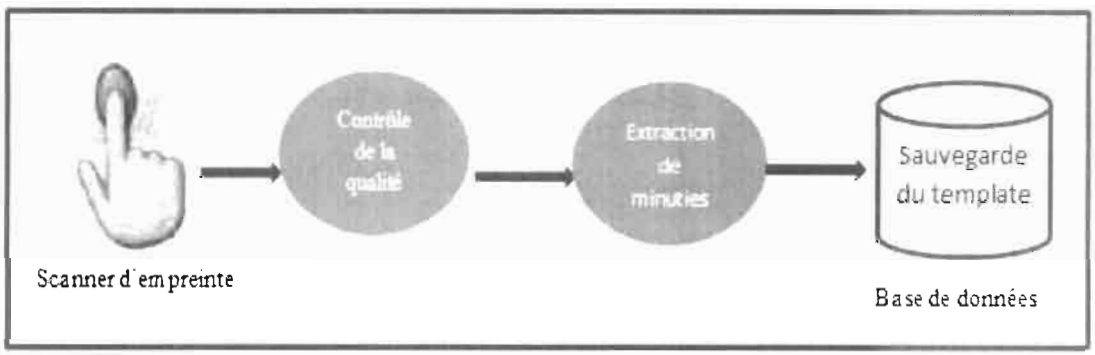

Figure II.5 : Processus d'enrôlement

La figure II.6 illustre le processus de vérification d'une empreinte digitale dans la base de données.

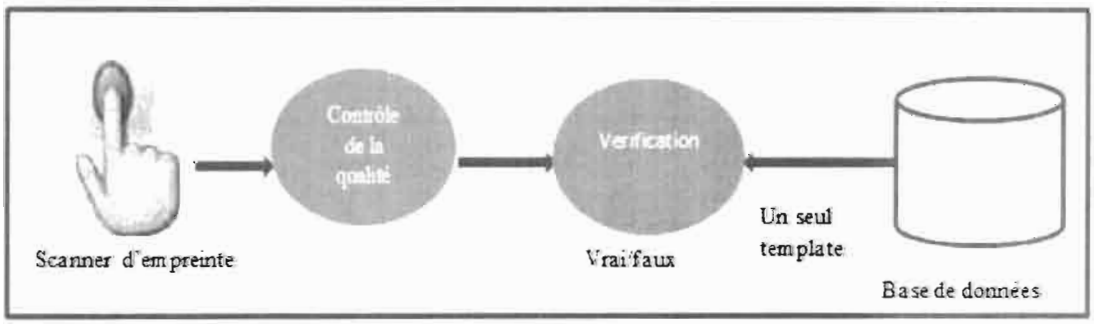

Figure II.6 : Processus de vérification

La figure II.7 illustre le processus d'identification d'une empreinte digitale dans la base de données.

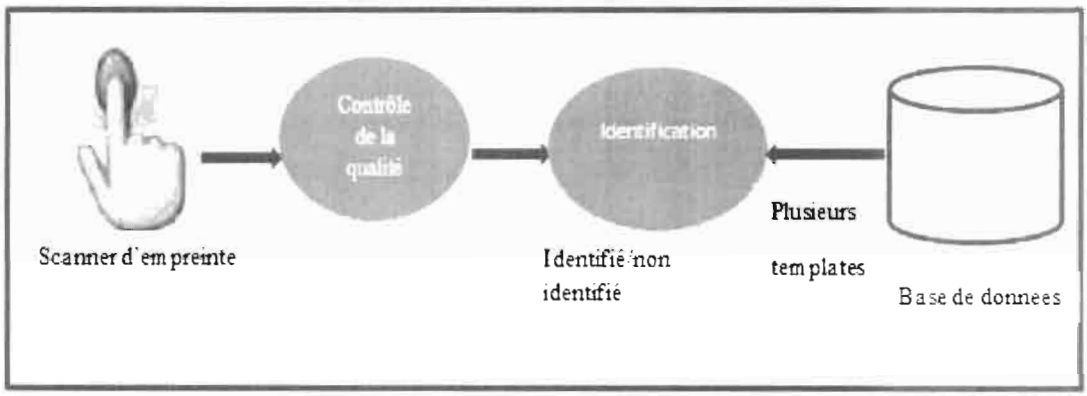

Figure II.7 : Processus d'identification

Conclusion

Nous avons pu définir dans ce chapitre un système biométrique et citer les différents types d'empreintes digitales. Nous avons vu également le traitement de l'empreinte digitale par les systèmes biométriques au cours d'une identification ou d'une vérification. Le chapitre suivant nous permettra d'entamer l'étude préliminaire.

Chapitre III : Etude préliminaire

Introduction

Dans cette partie, nous procéderons à l'analyse des besoins fonctionnels et non fonctionnels attendu de l'application qui doivent répondre à l'attente de l'utilisateur à travers la description des besoins du système. L'identification des besoins fonctionnels représente une étape importante du processus de développement 2TUP.

1. Les besoins fonctionnels

Les besoins fonctionnels listent les opérations réalisables de notre application. Ce sont des besoins spécifiant un comportement d'entrée / sortie du système. Le tableau III.1 donne la liste des différentes fonctionnalités attendues de l'application.

Tableau III.1 : Liste des besoins fonctionnels

N°	1	2	3	4	5	6	7
Fonctionnalités	Enrôler empreinte digitale	Identifier empreinte digitale	Vérifier empreinte digitale	Rechercher doublons	Authentifier	Gérer utilisateur	Administrer système

2. Les besoins opérationnels

Les besoins opérationnels représentent les besoins non fonctionnels, qui caractérisent le système comme la performance ainsi que la sécurité et l'ergonomie du système.

Ces besoins peuvent être énoncés suivant des plans de classifications.

- ❖ L'ergonomie
- L'interface doit être simple et claire : La manipulation de l'interface ne doit pas nécessiter des connaissances poussées,
- L'application doit être compatible avec n'importe quel système d'exploitation, facile à manipuler, compréhensible.

- ❖ Robustesse

L'application doit permettre le stockage d'un nombre maximal d'empreintes digitales, ainsi qu'une bonne gestion d'erreurs des images.

- ❖ Sécurité

L'application doit garantir à l'utilisateur connecté l'intégrité et la confidentialité de ses données. Notre système doit également certifier la disponibilité qui s'avère primordiale pour un bon fonctionnement.

- ❖ L'application doit garantir :
- fiabilité ;

- rapidité lors de l'identification.

3. Les choix techniques

❖ Architecture fonctionnelle/Module pour PC

Une architecture 3-tiers devra être utilisée pour permettre un déploiement facile de l'application. La figure III.1 illustre le fonctionnement d'une architecture 3-tiers, on y trouve 3 couches principalement :

- *une couche présentation se chargeant de l'affichage de l'interface utilisateur ;*
- *une couche intermédiaire assurant le dialogue ente l'interface utilisateur et la base de données ;*
- *la dernière couche est celle qui se charge du stockage des données.*

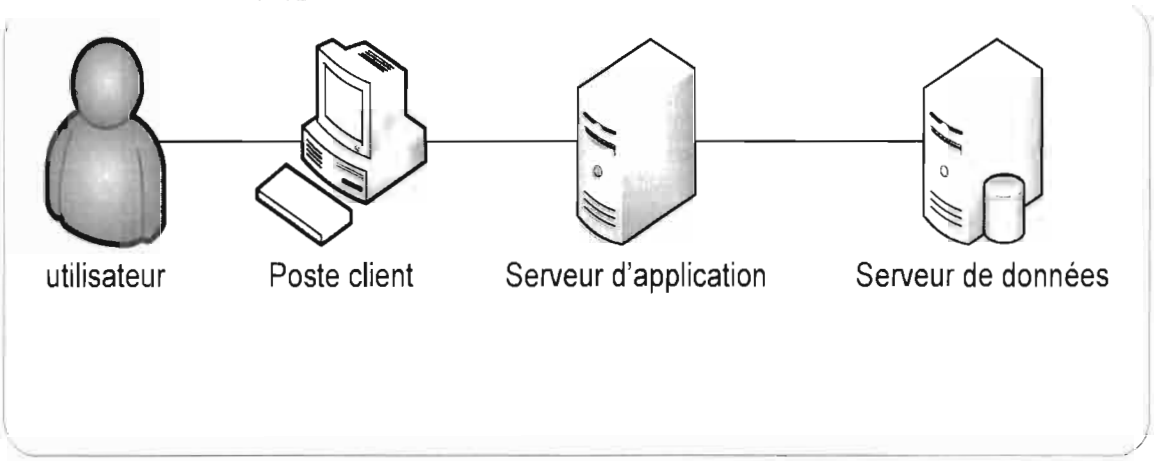

Figure III.1 : Architecture fonctionnelle/Module pour PC

❖ Architecture fonctionnelle/Module pour Android

Les services web seront utilisés pour le développement de la partie métier de notre application sur Android. L'architecture fonctionnelle sera composée également de 3 couches (figure III.2) notamment :

- *une couche présentation se chargeant de l'affichage de l'interface utilisateur sur les différentes tablettes ;*
- *une couche intermédiaire appelé middleware assurant le dialogue ente l'interface utilisateur et la base de données ;*
- *la dernière couche est celle qui se charge du stockage des données.*

Figure III.2 : Architecture de fonctionnelle/Module pour Android

❖ Architecture logicielle

Quant à l'architecture logicielle nous préconisons le modèle MVC pour une programmation professionnelle et l'assurance d'une maintenance facile de l'application. Le modèle se compose de trois grandes parties :

- *un modèle (modèle de données) ;*
- *une vue (présentation, interface utilisateur) ;*
- *un contrôleur (logique de contrôle, gestion des événements, synchronisation).*

La figure III.3 présente les différentes interactions entre les trois parties du modèle MVC.

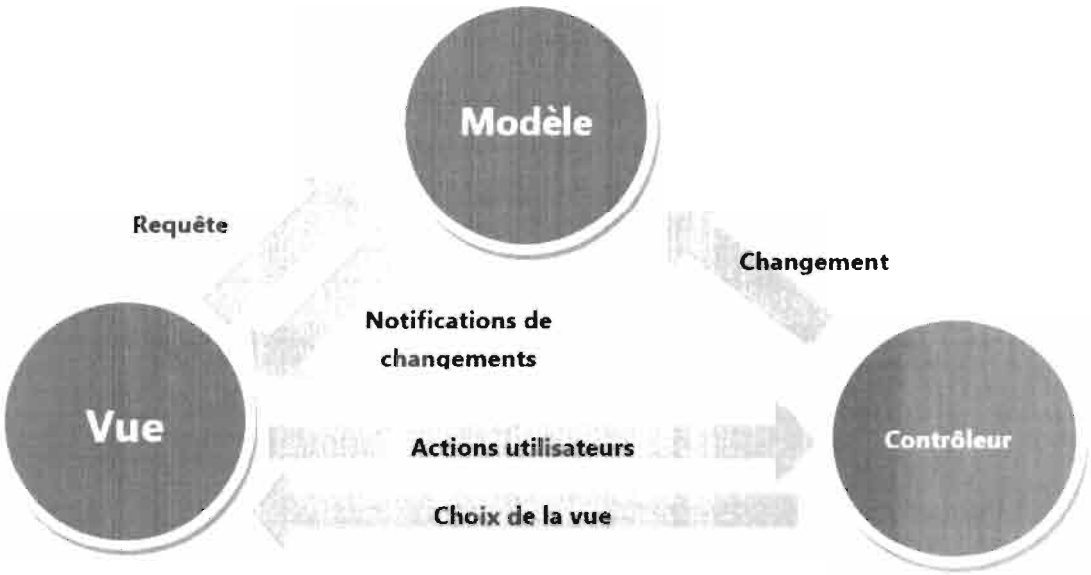

Figure III.3 : Modèle MVC

❖ Besoins techniques

Pour la réalisation du projet nous avons utilisé comme :

- SDK : **MegaMatcher**
- Environnement de développement : **Eclipse**

- Langage de programmation : **JAVA**
- Serveur d'application : **Apache-Tomcat**
- SGBD : **Oracle 10g**

Conclusion

Après avoir dégagé les besoins fonctionnels et opérationnels et tous les éléments qu'on doit prendre en considération pour l'analyse, on va entamer le chapitre suivant, par la formalisation des différents besoins. Ce chapitre nous permettra d'étayer les différents besoins attendus du système.

Chapitre IV : Capture des besoins du système

Introduction

Il s'agit dans cette étude de préciser le contexte fonctionnel du système. Plus spécifiquement, il s'agit de préciser la manière dont les acteurs devront utiliser le système. Cela se fera à travers l'identification des fonctionnalités du système (cas d'utilisations) ainsi que la représentation sous forme graphique (diagramme de cas d'utilisation et séquence) des interactions entre les acteurs et le système.

I. Capture des besoins fonctionnels

1. Identification des fonctionnalités (Cas d'utilisations)

Les cas d'utilisations permettent d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système. Aussi, pour identifier les cas d'utilisations, il faut se placer du point de vue de chaque acteur et déterminer comment et surtout pourquoi il se sert du système. Les différentes fonctionnalités sont citées dans le tableau IV.1.

Tableau IV.1 : Tableau descriptif des différentes fonctionnalités

N°	Titre	Description	Acteurs	Messages émis/reçus par les acteurs
1	Enrôler empreinte digitale	Permet de recueillir l'empreinte digitale ainsi que les informations relatives à une personne	Agents	Emet : Empreinte(s) Reçus : Confirmation
2	Identifier empreinte digitale	Permet d'identifier un individu à travers son empreinte digitale	Agents	Emet : Empreinte(s) Reçu : fiche de la personne identifiée
3	Vérifier empreinte digitale	Permet de vérifier la présence d'un individu à travers son empreinte digitale dans la base de données	Agents	Emet : Empreinte(s) Reçu : fiche de la personne vérifiée
4	Rechercher doublons	Permet de voir les empreintes en double dans la base de données	Agents	Emet : Date et heure Reçus : Liste des empreintes en double
5	Authentifier	Permet à un utilisateur d'avoir accès à l'application	Agents, Administrateur	Emet : Login et mot de passe Reçu : Confirmation
6	Gérer utilisateur	Permet de gérer les différents utilisateurs	Administrateur	Emet : Infos utilisateurs Reçu : Confirmation
7	Administrer système	Permet d'administrer le système	Administrateur	Emet : Infos système Reçu : Confirmation

2. Le diagramme de cas d'utilisation

Les **diagrammes de cas d'utilisation** sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. La figure IV.1 nous présente le diagramme de cas d'utilisation.

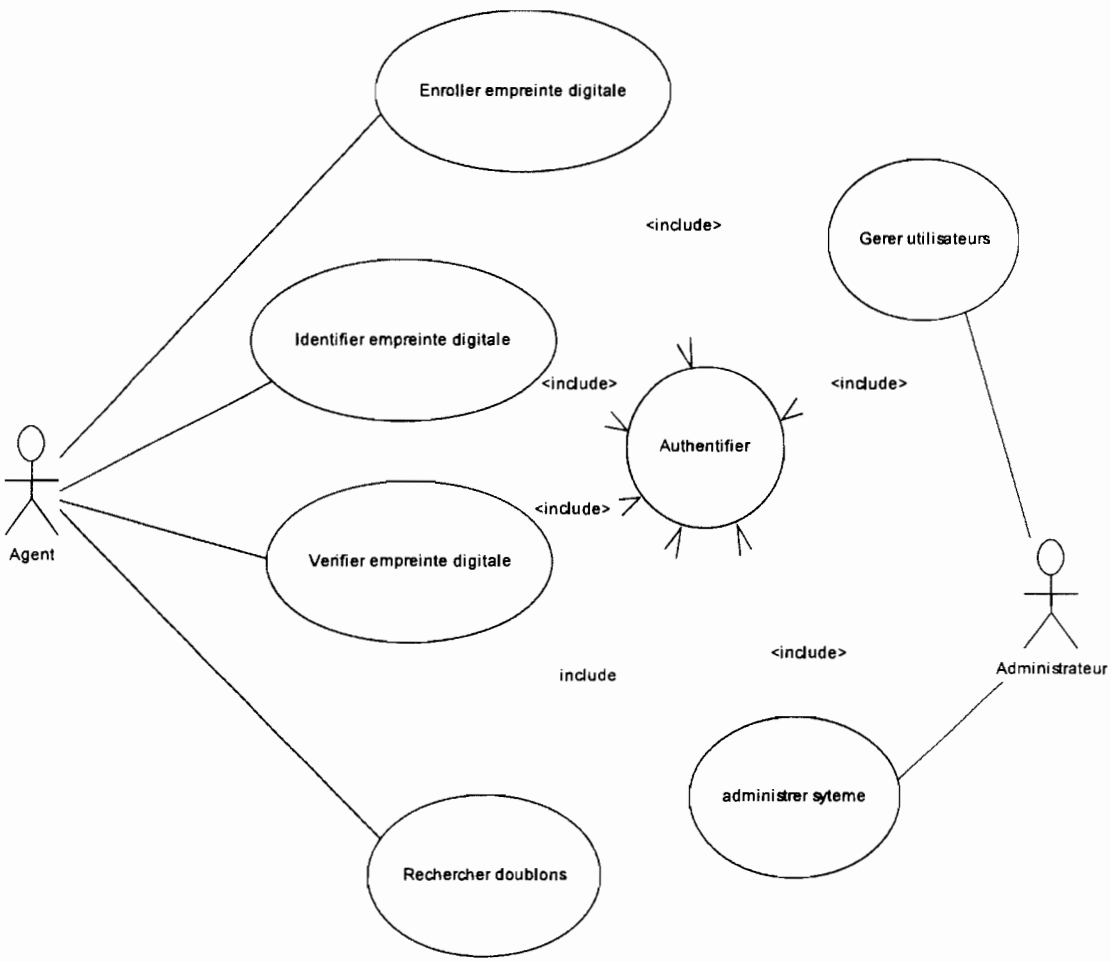

Figure IV.1 : Diagramme de cas d'utilisation

NB : Pour le module pour Android les cas d'utilisations suivants seront indispensables :

- rechercher doublons ;
- gestions utilisateurs ;
- administrer système.

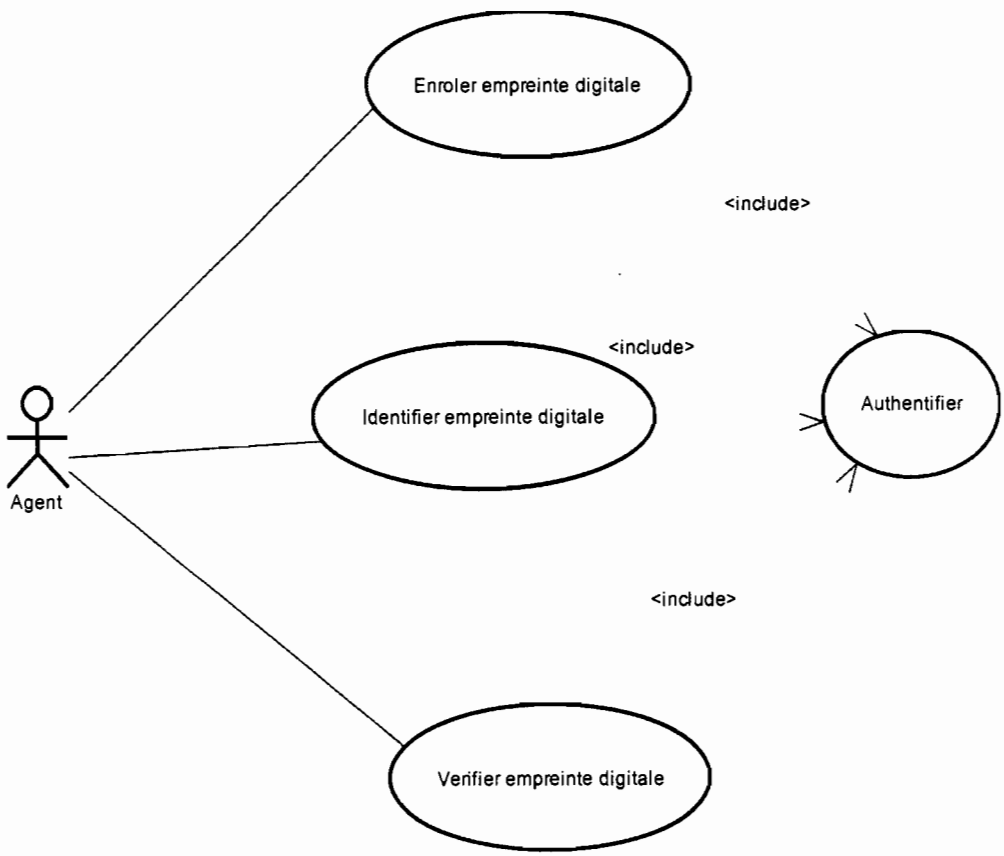

Figure IV.2 : Diagramme de cas d'utilisation/Module pour Android

3. Description détaillée des cas d'utilisation

a. *Enrôler empreinte digitale*

Les tableaux ci-dessous donnent une description textuelle de l'enrôlement d'une empreinte digitale. Le scenario nominal et le scenario alternatif sont représentés respectivement par le tableau IV.2a et IV.2b.

Tableau IV.2a : Enrôler empreinte digitale

Description sommaire			
Titre : Enrôler empreinte digitale	***A***cteurs		***V***ersion : 1.0
	Principaux : Agents	***S***econdaire :	
But : Permet de recueillir l'empreinte digitale ainsi que les informations relatives à une personne			

Description des enchainements

Pré-Condition

Utilisateur connecté

« « début » »

Scénario Nominal

Utilisateur	***S***ystème
01. Demande à enrôler individu	02. Affiche un formulaire
03. Saisit les informations demandées puis clique sur «suivant »	04. Vérifie les informations fournies
06. Insère empreinte digitale et clique sur « suivant »	05. Demande renseignement de l'empreinte digitale
	07. Vérifie Unicité de l'empreinte
	08. Valide les informations et affiche message de réussite de l'enregistrement

Tableau IV.2b : Enrôler empreinte digitale

*S*cénario *A*lternatif

1A. Les informations du formulaire sont incorrectes

*U*tilisateur	*S*ystème
	1A1. Affiche message d'erreur Le scénario reprend à l'étape 02

1B. L'empreinte renseignée est de mauvaise qualité

*U*tilisateur	*S*ystème
	1B1. Affiche message d'erreur Le scénario reprend à l'étape 05

1C. Unicité de l'empreinte non vérifié

*U*tilisateur	*S*ystème
	1C1. Affiche fiche de la personne correspondante à l'empreinte « « Fin » »

*P*ost-*C*ondition

Un enrôlement effectué

« « Fin » »

Le digramme de séquence du cas d'utilisation «Enrôler empreinte digitale» représenté par la figure IV.3.

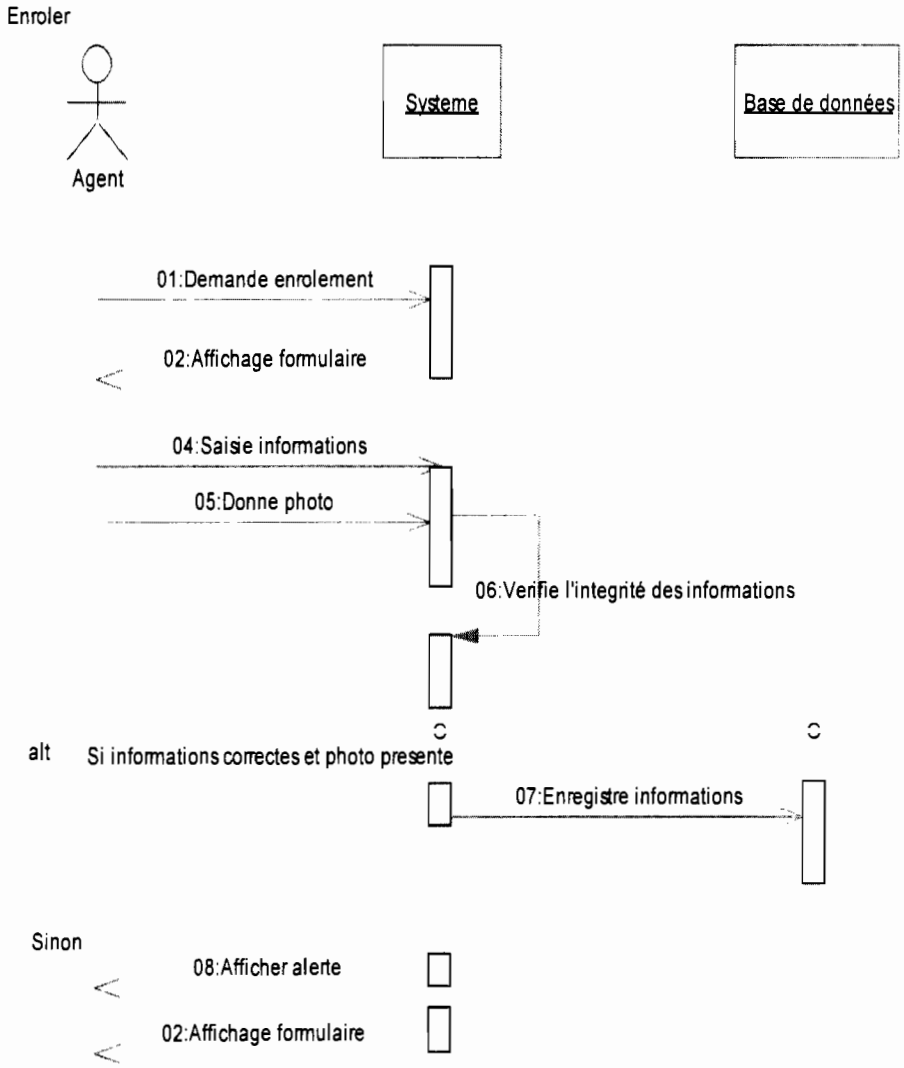

Figure IV.3 : Diagramme de séquence : Enrôler empreinte

b. Identifier empreinte digitale

Les tableaux ci-dessous donnent une description textuelle de l'identification d'une empreinte digitale. Le scenario nominal et le scenario alternatif sont représentés respectivement par le tableau IV.3a et IV.3b.

Tableau IV.3a : Identifier empreinte digitale

Description sommaire			
Titre : Identifier empreinte digitale	***A***cteurs		***V***ersion : 1.0
	Principaux : Agents	***S***econdaire :	
But : Permet d'effectuer l'identification d'une personne			
Description des enchainements			
Pré-Condition			
Utilisateur connecté			
« « début » »			
Scénario Nominal			
Utilisateur	***S***ystème		
01. Demande à identifier individu 03. Insère empreinte digitale et clique sur « suivant »	02. Demande renseignement de/ou des empreintes digitale(s) 04. Vérifie les empreintes fournies 05. Valide les informations et affiche fiche de la personne identifiée		

Tableau IV.3b : Identifier empreinte digitale

*S*cénario *A*lternatif	
1A. Les informations du formulaire sont incorrectes	
*U*tilisateur	*S*ystème
	1A1. Affiche message d'erreur Le scénario reprend à l'étape 02
1B. L'empreinte renseignée est de mauvaise qualité	
*U*tilisateur	*S*ystème
	1B1. Affiche message d'erreur Le scénario reprend à l'étape 02
1C. Individu non identifié	
*U*tilisateur	*S*ystème
	1C1. Affiche message « « Fin » »
*P*ost-*C*ondition Une identification effectuée ***« « Fin » »***	

Le digramme de séquence du cas d'utilisation «Identifier empreinte digitale» représenté par la figure IV.4.

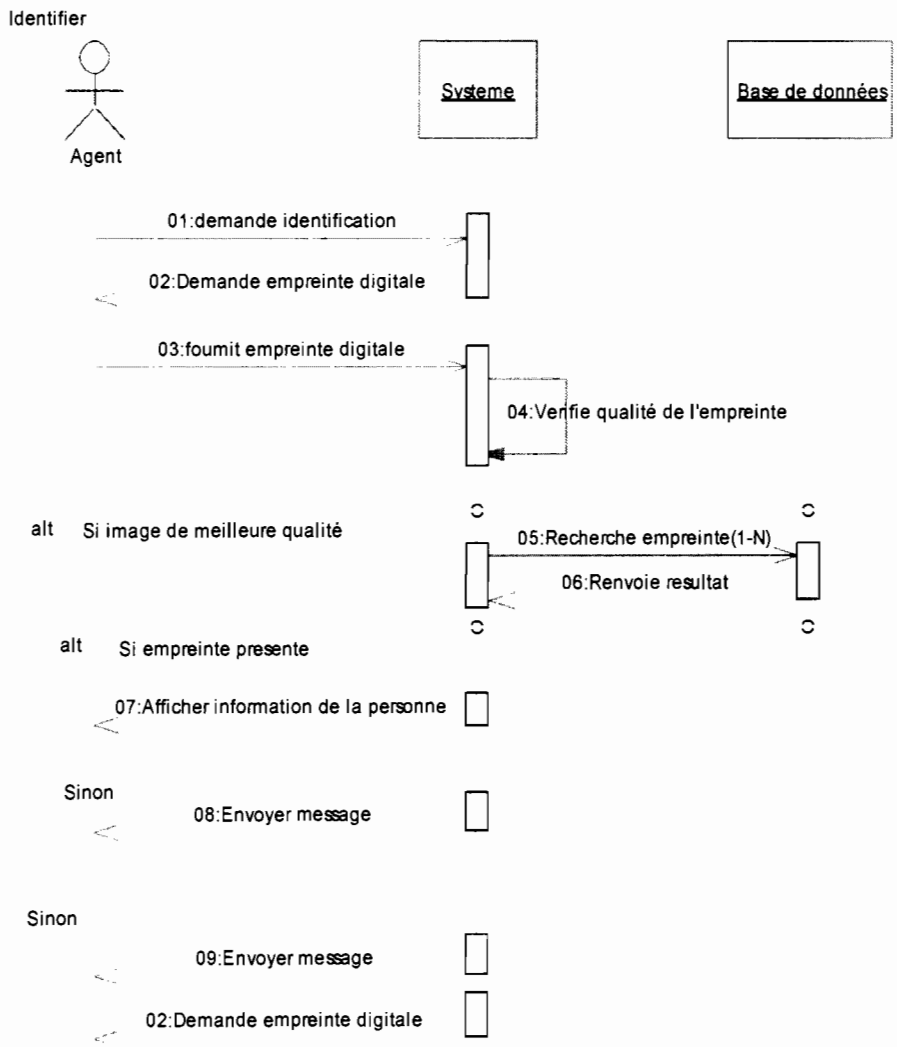

Figure IV.4 : Diagramme de séquence : Identifier empreinte digitale

c. Vérifier empreinte digitale

Les tableaux ci-dessous donnent une description textuelle de la vérification d'une empreinte digitale. Le scenario nominal et le scenario alternatif sont représentés respectivement par le tableau IV.4a et IV.4b.

Tableau IV.4a : Vérifier empreinte digitale

Description sommaire

Titre : Vérifier empreinte digitale	Acteurs		Version : 1.0
	Principaux : Agents	Secondaire :	

But : Permet de vérifier la présence d'un individu à travers son empreinte digitale dans la base de données

Description des enchainements

Pré-Condition

Utilisateur connecté

« « début » »

Scénario Nominal

Utilisateur	Système
01. Demande à effectuer vérification	02. Affiche page de vérification
03. Renseigne empreinte digitale	04. Vérifie validité de l'empreinte
	05. Demande de choisir empreinte digitale à vérifier
	06. Affiche message de correspondance entre les deux empreintes

Tableau IV.4b : Vérifier empreinte digitale

*S*cénario *A*lternatif	
***1A.* L'empreinte renseignée est de mauvaise qualité**	
*U*tilisateur	*S*ystème
	1A1.Affiche message d'erreur Le scénario reprend à l'étape 02
1B. Non correspondance des deux empreintes digitales	
*U*tilisateur	*S*ystème
	1B1. Affiche message de non correspondance *« « **Fin** » »*
*P*ost-*C*ondition Empreinte vérifiée *« « **Fin** » »*	

Le digramme de séquence du cas d'utilisation «Vérifier empreinte digitale» représenté par la figure IV.5.

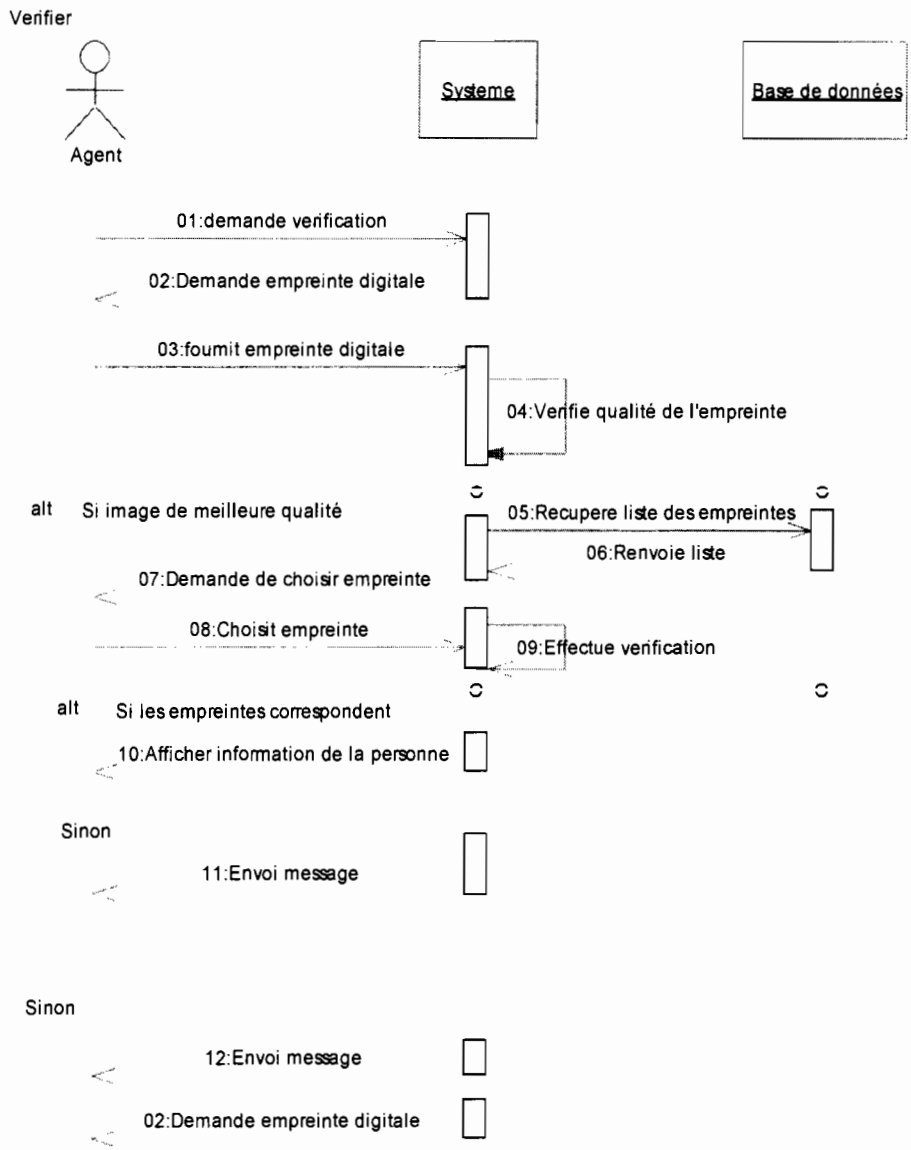

Figure IV.5 : Diagramme de séquence : Vérifier empreinte digitale

d. Rechercher doublons

Les tableaux ci-dessous donnent une description textuelle de la recherche de doublons dans la base de données. Le scenario nominal et le scenario alternatif sont représentés respectivement par le tableau IV.5a et IV.5b.

Tableau IV.5a : Rechercher doublons

Description sommaire

*T*itre : Rechercher doublons	*A*cteurs		*V*ersion : 1.0
	*P*rincipaux : Agents	*S*econdaire :	

*B*ut : Permet de voir les empreintes en double dans la base de données

Description des enchainements

Pré-Condition

Utilisateur connecté

« « début » »

Scénario Nominal

*U*tilisateur	*S*ystème
01. Demande programmation horaire de vérification	02. Affiche page de programmation
03. Programme l'heure de vérifications	04. Effectue recherche automatique de doublons
	05. Affiche les doublons

Tableau IV.5b : Rechercher doublons

Scénario ***A***lternatif	
1A. Absence de doublons	
Utilisateur	***S***ystème
	1A1. Affiche Message d'absence ***« « Fin » »***
Post-***C***ondition Vérification effectuée ***« « Fin » »***	

Le digramme de séquence du cas d'utilisation «Rechercher doublons» représenté par la figure IV.6.

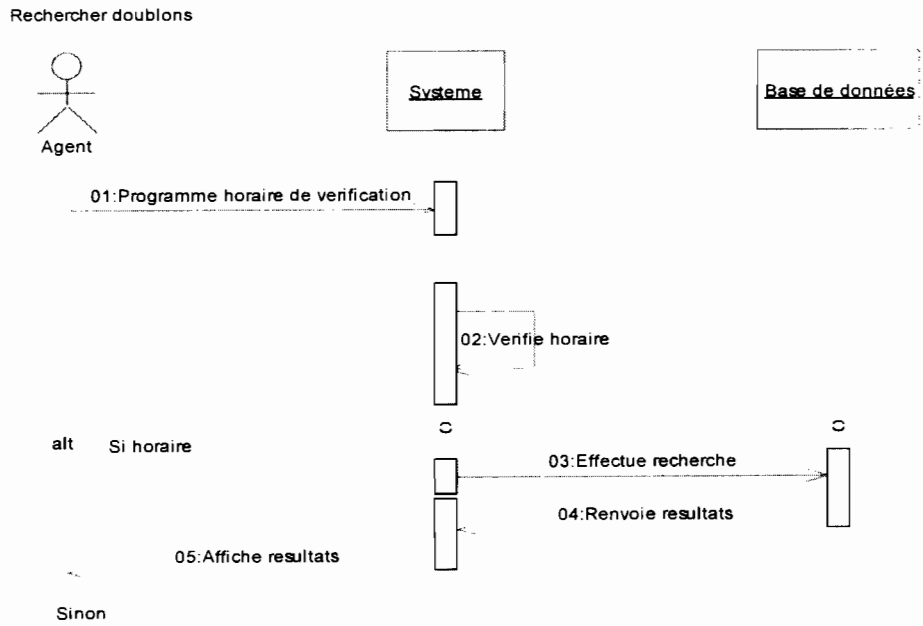

Figure IV.6 : Diagramme de séquence: Rechercher doublons

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

e. S'authentifier

Les tableaux ci-dessous donnent une description textuelle du cas d'utilisation «s'authentifier». Le scenario nominal et le scenario alternatif sont représentés respectivement par le tableau IV.6.

Tableau IV.6 : S'authentifier

Description sommaire

Titre : S'authentifier	Acteurs		Version : 1.0
	Principaux : Tous les utilisateurs	Secondaire :	

But : Permet à un utilisateur d'avoir accès à l'application

Description des enchainements

Pré-Condition

Utilisateur déjà enregistré

« « début » »

Scénario Nominal

Utilisateur	Système
01. Demande à accéder à l'application	02. Affiche un formulaire
03. Saisit les informations demandées puis clique sur « connecter »	04. Vérifie les informations fournies
	05. Valide les informations fournies
	06. Affiche l'environnement de travail de l'utilisateur

Scénario Alternatif

1A. Login et/ou mot de passe incorrect(s)

Utilisateur	Système
	1A1.Affiche message d'erreur
	Le scénario reprend à l'étape 02

Post-Condition

Un Utilisateur connecté ***« « Fin » »***

Le digramme de séquence du cas d'utilisation «S'authentifier» représenté par la figure IV.7.

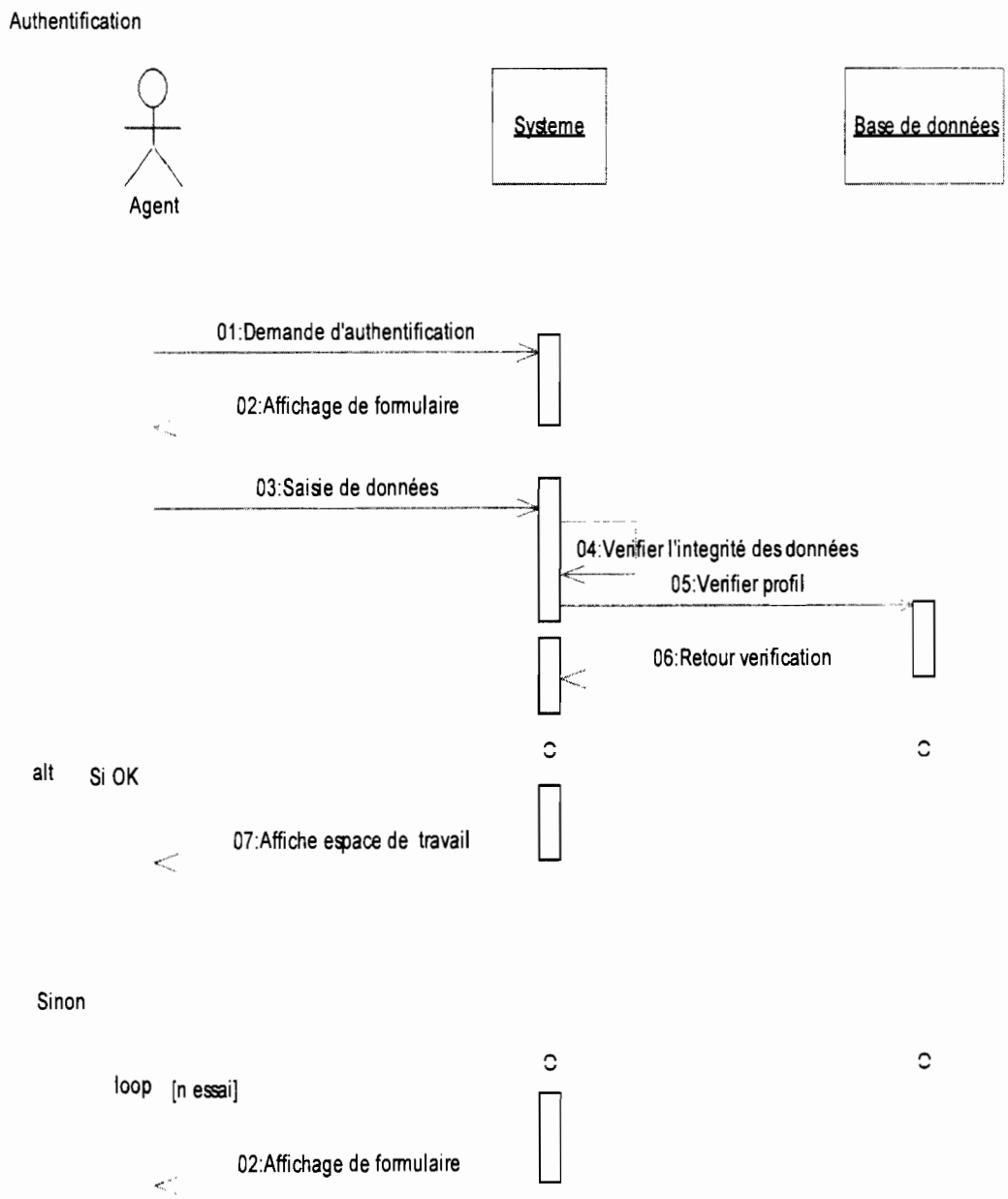

Figure IV.7 : Diagramme de séquences : S'authentifier

II. Capture des besoins techniques

1. Présentation du SDK MegaMatcher

MegaMatcher est un SDK(*Software Development Kit* ou *kits de développement logiciels en français*) biométrique développé par la société Neurotechnology spécialisé dans le développement d'algorithmes et de produits de développement de logiciel pour les empreintes digitales, visage, iris, reconnaissance de forme, la vision par ordinateur et de reconnaissance d'objets. L'architecture fonctionnelle su SDK est présentée par la figure IV.8.

MegaMatcher fournit :

- des algorithmes de traitement d'image (segmentation, extraction),
- un moteur de recherche rapide et fiable.

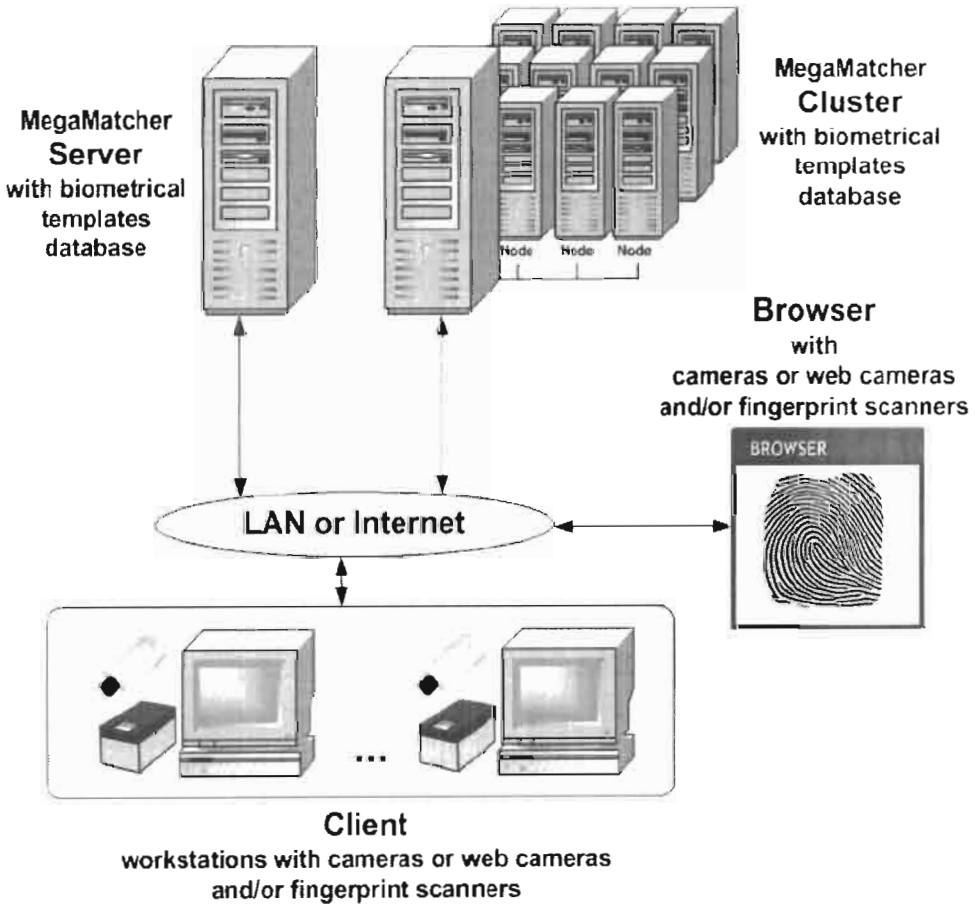

Figure IV.8 : Architecture technique du SDK

2. L'architecture de déploiement

Les systèmes à développer doivent être accessibles à distance avec une base de données centralisée, alors une architecture 3-tiers a été choisie pour le déploiement des deux applications. La figure IV.9 et la figure IV.10 Illustrent respectivement l'architecture 3-tiers prévu pour le déploiement de l'application sur PC et sur terminaux mobiles.

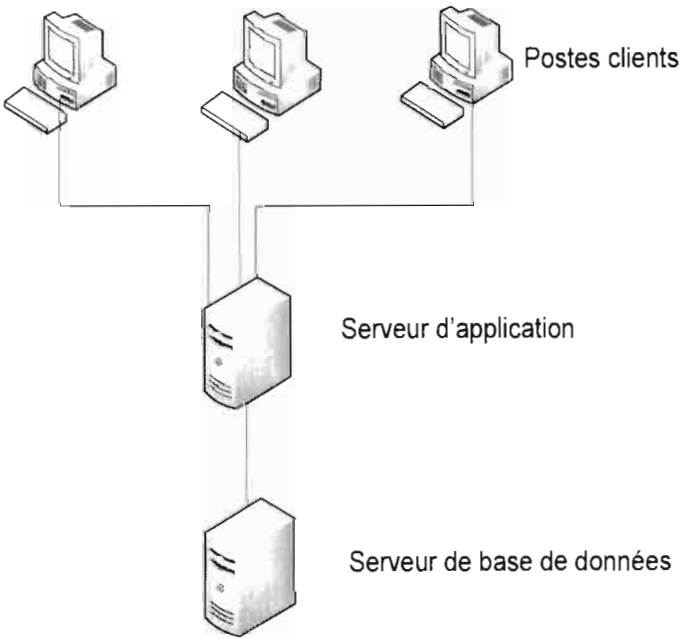

Figure IV.9 : Architecture de déploiement/module pour PC

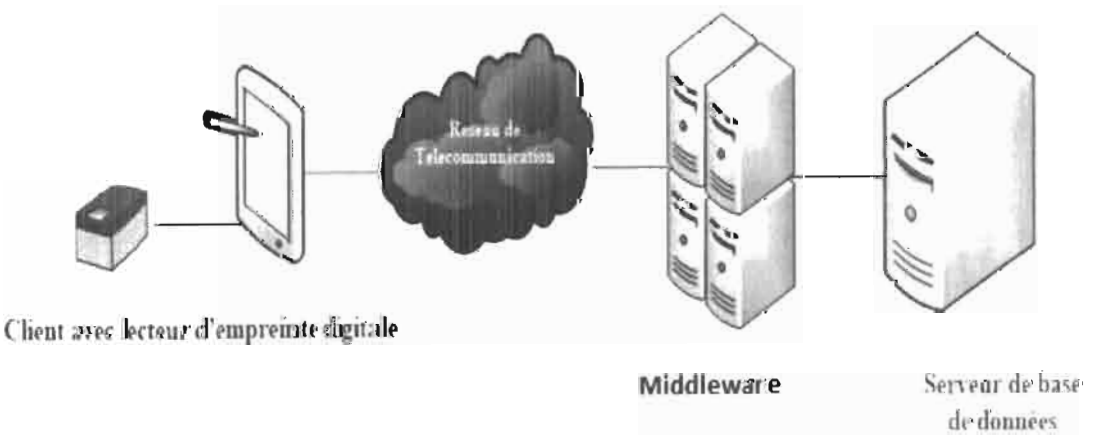

Figure IV.10 : architecture de déploiement/module pour Android

3. Les architectures réseaux

L'architecture réseau présente le schéma du réseau (voir figure IV.11) et l'ensemble du matériel informatique nécessaire et le protocole de communication mis en place entre le serveur de données et les clients pour une bonne utilisation de l'application durant la phase de production. La figure IV.12 figure IV.13 représentent respectivement l'architecture réseau du module pour PC et pour Android.

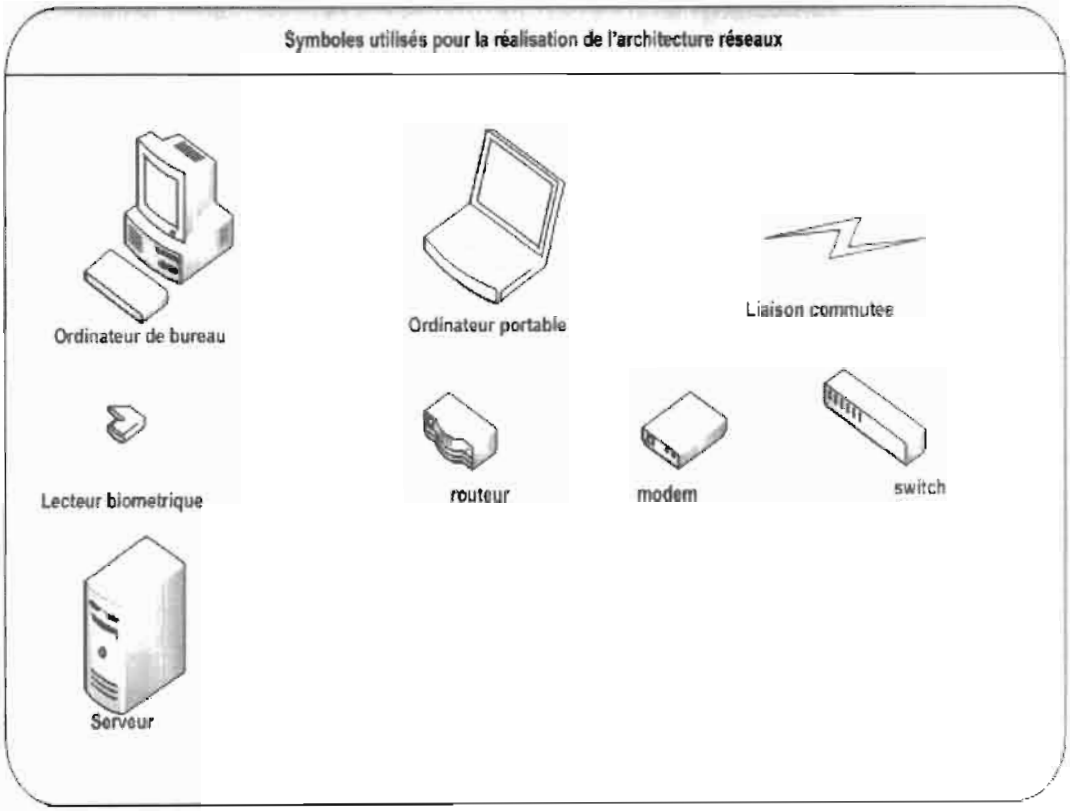

Figure IV.11: Symboles /architecture réseau

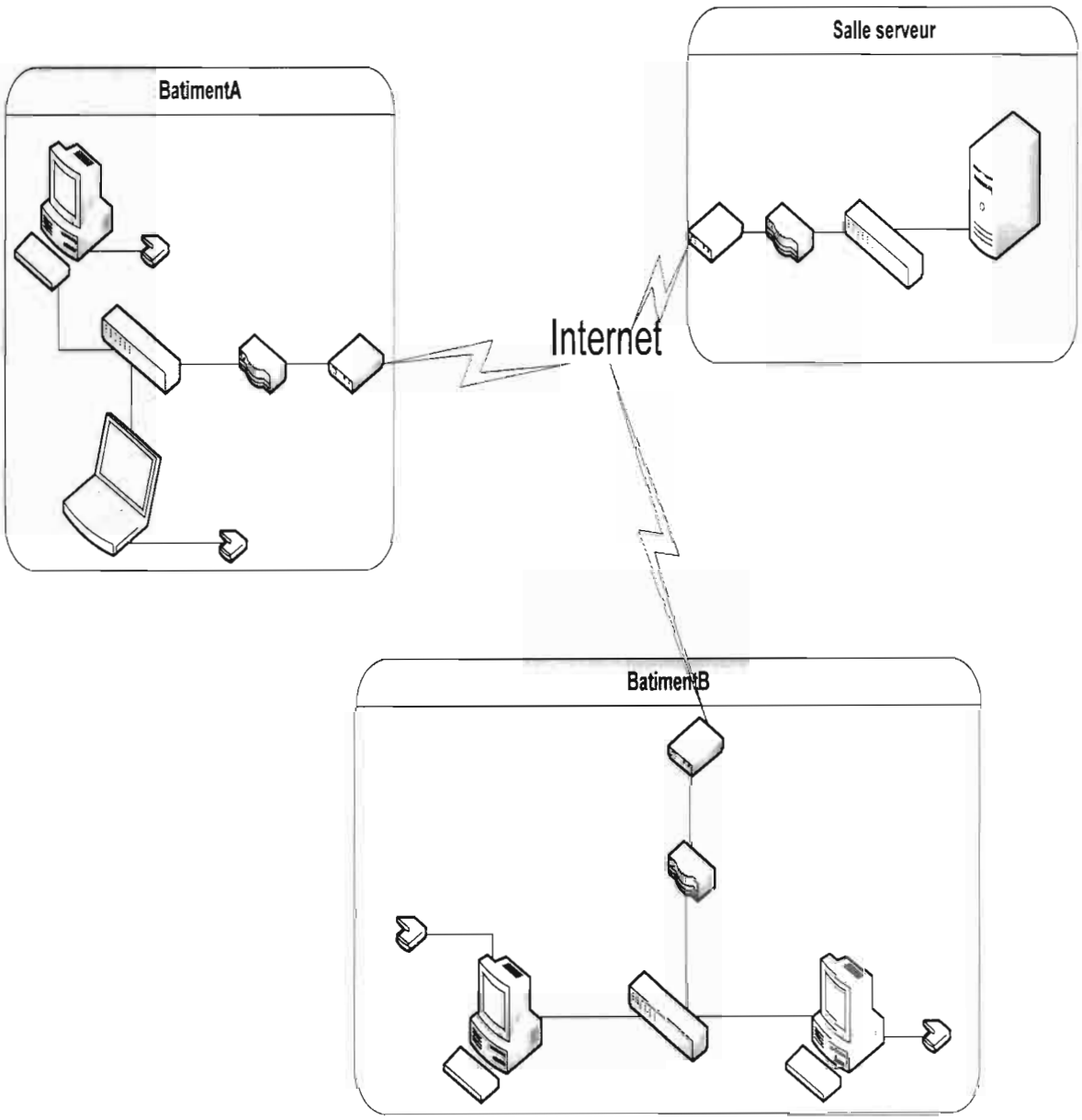

Figure IV.12 : Architecture réseau/Module pour PC

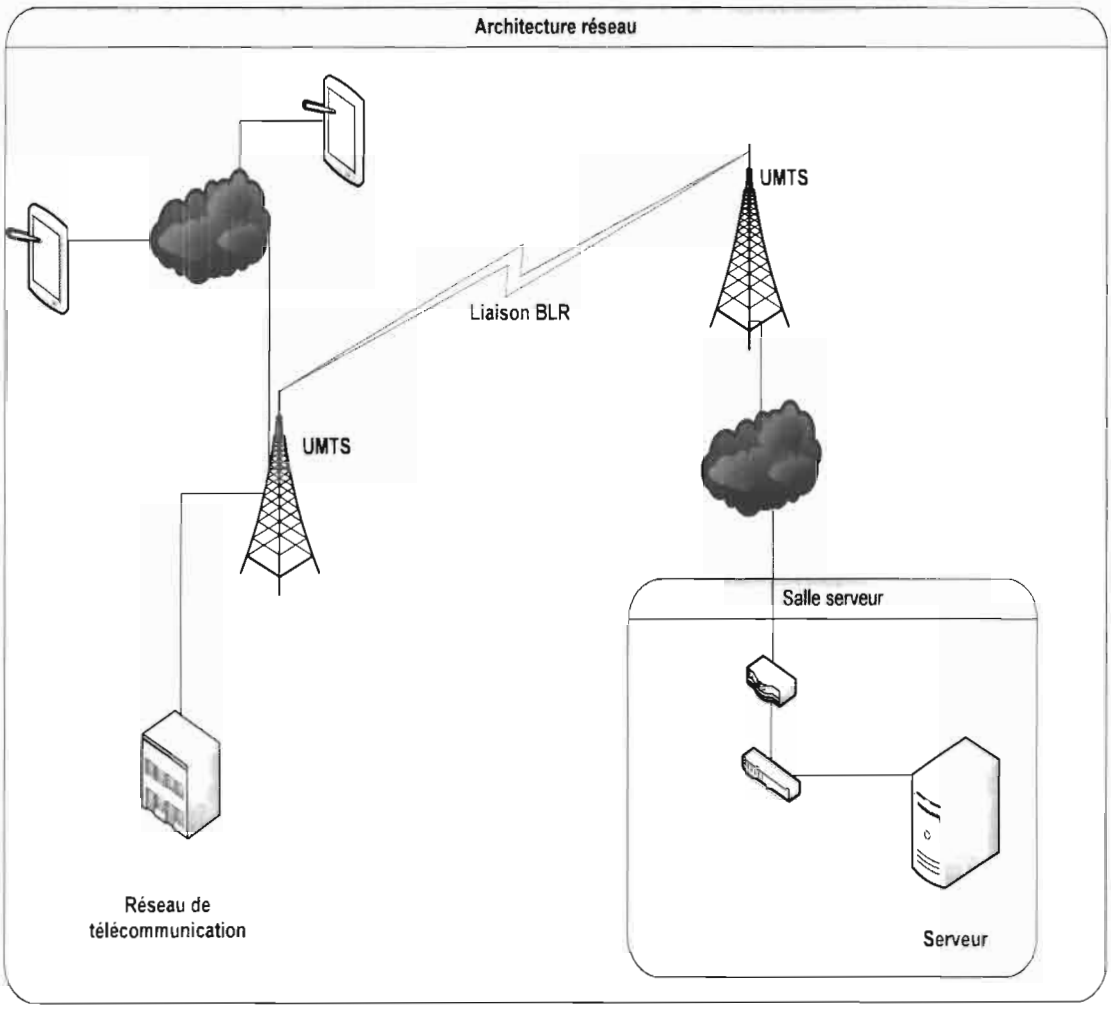

Figure IV.13 : Architecture réseau/Module pour Android

Les agents munis de leurs tablettes et en présence du réseau de télécommunication devront envoyer les informations au réseau de télécommunication qui se chargera d'acheminer ces données à travers une connexion BLR vers une antenne dédiée qui à son tour transmettra les données au serveur. La réponse du serveur suit le chemin inverse et l'agent reçoit ainsi la réponse sur sa tablette.

4. Coût de mise en œuvre

a. *Besoins logiciels*

Pour la partie conception et réalisation un certain nombre de logiciels devraient être utilisé. L'ensemble des logiciels est présenté dans le tableau IV.7.

Tableau IV.7 : Liste des besoins logiciels/mise en œuvre

Désignation	Caractéristiques	Prix unitaire (FCFA)	Quantité
IDE	Eclipse juno	gratuit	1
Logiciel de modélisation	Power AMC	Existant	1
SDK	Android SDK	gratuit	1
SDK	Fingerprint SDK(VeryFinger Extended SDK)	562 645	1
Apache Tomcat		gratuit	1
Base de données	Oracle10g	existant	1

b. *Besoins matériels*

Le tableau IV.8 contient les différents matériels à utiliser pendant la phase de mise en œuvre du projet.

Tableau IV.8 : Liste des besoins matériels/mise en œuvre

Désignation	Caractéristiques	Prix unitaire (FCFA)	Quantité
PC	-Ram : 1 Go ou plus. -Processeur : Au moins 2 GHz. -Disques dur : 100 G -Système d'exploitation : Windows 7	existant	1
Tablette	Marque : Samsung OS : Android	225 000	1
Lecteur d'empreinte digital	Futronic FS50	153 270	1
Webcam	Logitech HD Pro C910	72 181	1

c. Coût de mise en œuvre

Tableau IV.9 : Coût de mise en œuvre

Besoins	Coûts (FCFA)
logiciels	562 645
Matériels	225 451
Total	788 096

5. Coût de mise en production

a. Besoins logiciels

La mise en production requiert l'achat du SDK dont le prix est fonction du nombre de poste (voir tableau IV.10a), les autres logiciels nécessaires sont énumérés dans le tableau IV.10b.

Tableau IV.10a : Liste besoins logiciels/production

Nombre de PC	Fingerprint Client(FCFA)	Fingerprint Matcher(FCFA)	Total (FCFA)
1 - 9	24 500	14 500	39 300

Tableau IV.10b : Liste besoins logiciels/production

Désignation	Caractéristiques	Prix unitaire (FCFA)	Quantité
Apache Tomcat	Version 6.0.39	gratuit	1
Base de données	Oracle10g	existant	1
OS (serveur d'application)	Linux Debian	gratuit	1

b. Besoins matériels minimal

Le tableau IV.11 contient les différents matériels à utiliser pendant la phase de production ainsi que leur nombre.

NB : Pour les besoins matériels, nous allons partir sur le principe qu'on dispose de 9 postes clients.

Tableau IV.11 : Liste matériel minimal/production

Désignation	Caractéristiques	Prix unitaire(FCFA)	Quantité
Serveur (application et base de données)	-Ram : 1 Go ou plus. -Processeur : Au moins 2 GHz. -Disques dur : 100 G -Système d'exploitation : Linux Debian -Sun Java 2 -Composants VeriFinger Extended	756 000	1
Lecteur d'empreinte digital	Futronic FS50	163 800	9
Tablette	Marque : Samsung OS : Android	225 000	9
Webcam	Logitech HD Pro C910	72 181	9

c. *Coût de production minimal*

Tableau IV.12 : Cout de production minimal

Besoins	Coûts (FCFA)
logiciels	39 300
matériels	2 879 829
TOTAL	2 919 129

Conclusion

Dans ce chapitre nous avons identifié les différentes fonctionnalités attendues (cas d'utilisations) ainsi que la représentation sous forme graphique de ces fonctionnalités. Nous avons fait cas de la liste de matériels et logiciels nécessaires pour le développement ainsi que pour la production. Le prochain chapitre va nous permettre de voir les spécifications conceptuelles.

Chapitre V : Spécifications conceptuelles

Introduction

Dans ce chapitre, nous présenterons les différentes classes ainsi que les relations entre ces classes à travers un diagramme de classe. Une description textuelle de ces différentes classes est également donnée pour une meilleure perception des différents attributs composants les différentes classes.

1. Généralité

Le diagramme de classe représente le concept le plus important dans un développement orienté objet. Il représente la structure statique du système c'est à dire la partie du système qui sera toujours présente lors des interactions définies par les cas d'utilisation. Le diagramme de classe permet donc de représenter les structures des classes composant le système et les relations entre les classes. La figure V.1 donne une présentation du diagramme de classe de notre système.

2. Le diagramme de classe

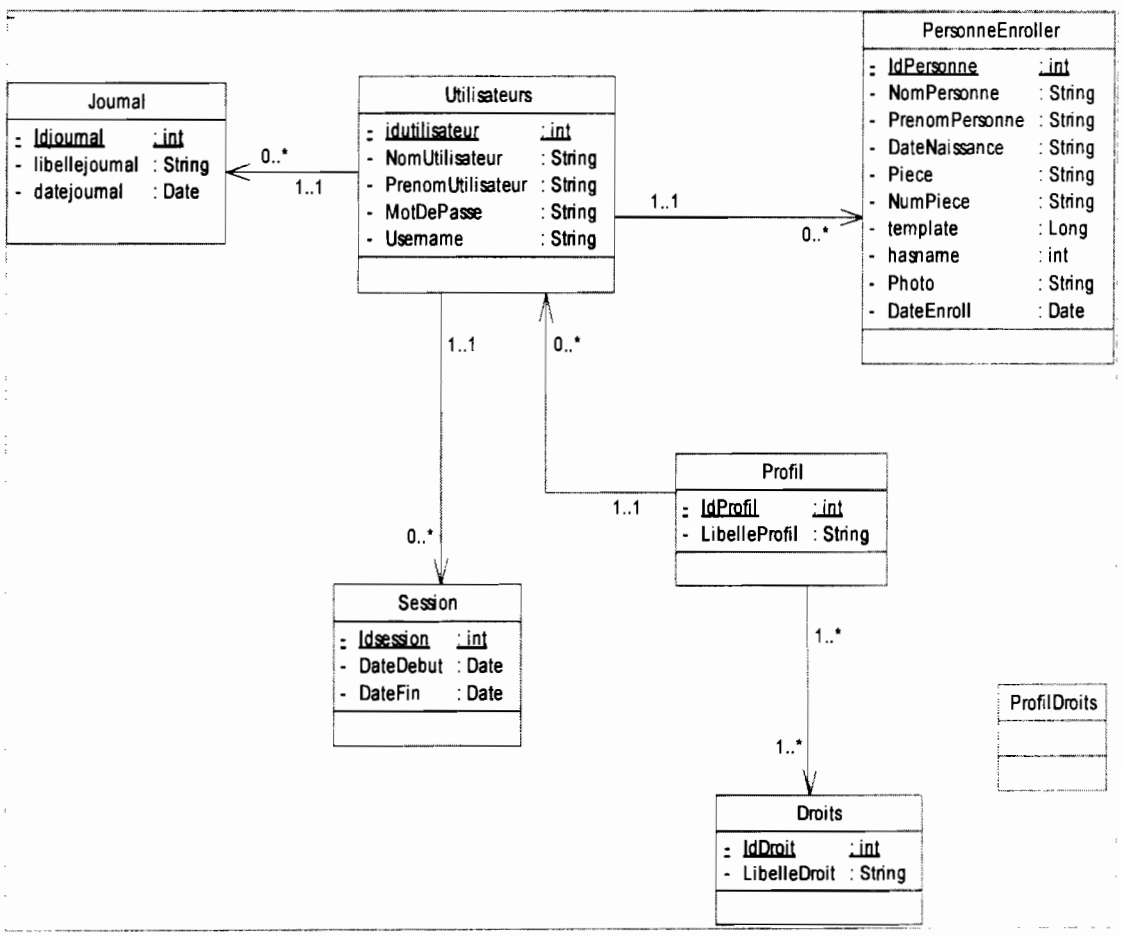

Figure V.1 : Diagramme de classes

3. Description textuelle du diagramme de classe

Pour une meilleure perception des différents attributs composants les différentes classes une description textuelle de ces attributs est nécessaire. Ces descriptions concernent notamment les classes « PersonneEnroler » (voir tableau V.1), « Utilisateurs » (voir tableau V.2), « Journal » (voir tableau V.3), « Session » (voir tableau V.4), « Profils » (voir tableau V.5), « Droits » (voir tableau V.6).

Tableau V.1 : PersonneEnroler

Classe : PersonneEnroler			**Permet d'enregistrer toutes les personnes ainsi que leurs empreintes**
Attributs			
Nom	Description	Type	Observation
IdPersonne	Identifiant d'une personne	TinyInt	Clé primaire
NomPersonne	Nom de la personne	Varchar(200)	
PrenomPersonne	Prénom de la personne	Varchar(200)	
DateNaissance	Date de naissance de la personne	Date	
Piece	Le type de pièce fournit par la personne	Varchar(100)	
NumPiece	Numéro de la pièce fournit	Varchar(100)	
Template	Ensemble des empreintes de la personne	Binaire	
Photo	Photo de la personne	Binaire	
DateEnroll	Date et heure de l'enrôlement	Date	

Tableau V.2 : Utilisateurs

Classe : Utilisateurs			**Permet d'enregistrer les différents utilisateurs**
Attributs			
Nom	Description	Type	Observation
IdUtilisateur	Identifiant d'un utilisateur	TinyInt	Clé primaire
NomUtilisateur	Nom de l'utilisateur	Varchar(200)	
PrenomUtilisateur	Prénom de l'utilisateur	Varchar(200)	
MotDePasse	Mot de passe de l'utilisateur	Varchar(200)	
Username	Login de l'utilisateur	Varchar(200)	

Tableau V.3 : Journal

Classe : Journal			**Permet d'enregistrer les différentes actions effectuées par l'utilisateur**
Attributs			
Nom	Description	Type	Observation
IdJournal	Identifiant	TinyInt	Clé primaire
LibelleJournal	Libellé du journal	Varchar(200)	
DateJournal	Date et heure de l'action	Date	

Tableau V.4 : Session

Classe : Session	**Permet d'enregistrer les différentes heures de connexions des utilisateurs**

Attributs

Nom	Description	Type	Observation
IdSession	Identifiant	TinyInt	Clé primaire
DateDebut	Début de la connexion	Date	
Datefin	Fin de la connexion	Date	

Tableau V.5 : Profil

Classe : Profil	**Permet d'enregistrer les différents profils des utilisateurs**

Attributs

Nom	Description	Type	Observation
IdProfil	Identifiant	TinyInt	Clé primaire
LibelleProfil	Profil	Varchar(200)	

Tableau V.6 : Droits

Classe : Droits	**Permet d'enregistrer les différents droits relatifs aux profils**

Attributs

Nom	Description	Type	Observation
IdDroit	Identifiant	TinyInt	Clé primaire
LibelleDroit	Droit	Varchar(200)	

Conclusion

Nous avons présenté dans ce chapitre le diagramme de classe de notre système ainsi que la description des différentes classes. Le chapitre suivant nous permettra d'entamer la partie réalisation de notre processus de développement.

Chapitre VI : Réalisation

Introduction

Ce chapitre nous permettra de donner les différentes configurations du projet ainsi que les différents outils utilisés pour l'implémentation de la solution. Nous présenterons également la procédure de déploiement du module pour PC et finirons par une présentation de quelques écrans des deux applications réalisées.

I. Configurations/Module pour PC

1. Configurations des outils d'implémentations

Le projet est construit et géré par Eclipse et l'outil Apache Maven. Notons qu'Apache Maven est un outil pour la gestion et l'automatisation de production des projets logiciels java en général et Java EE en particulier. L'objectif recherché est de produire un logiciel à partir de ses sources (modules externes, bibliothèques), en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

a. Installation de Maven

Télécharger Eclipse juno en utilisant l'adresse suivante :
http://www.eclipse.org/downloads/download.php?

Télécharger apache-maven3.1.1 en suivant le lien suivant :
http://maven.apache.org/download.html et placez le dans C:\Program Files (x86)\

Pour installer le plugin Maven les étapes suivantes sont nécessaires :

- démarrer Eclipse;
- allez à Help ->Install new software;
- dans le champ "Work with" entrer l'adresse suivante :
  http://download.eclipse.org/technology/m2e/releases;
- cliquez sur le bouton "Finish" pour terminer avec l'assistant.

b. Importation du projet

Démarrer d'abord Eclipse et suivez les étapes suivantes :

- cliquez sur File -> Import;
- Maven -> Existing Maven Projects -> Next;
- dans le champ "Root director" entrez "chemin_SDK\Neurotec_Biometric_4_3_SDK_Trial\Tutorials" (or PATH_TO_SDK\Neurotec_Biometric_4_3_SDK_Trial\Samples) ;
- sélectionnez les packages que vous voulez;
- cliquez sur le bouton "Finish" ;
- clic droit sur le projet exporté Run As -> Maven Install. Attendez que la construction soit finie avec succès;
- clic droit sur le projet exporté Debug As -> Debug Configurations;
- cliquez sur l'onglet "Argument";
- cliquez sur "File System" et spécifiez le chemin suivant chemin_SDK \Neurotec_Biometric_4_3_SDK_Trial\Bin\Win32_x86 si le JVM est de 32 bits ou PATH_TO_SDK\Neurotec_Biometric_4_3_SDK_Trial\Bin\Win64_x64 si le JVM est de 64 bits;
- cliquez sur l'onglet "Main" et spécifiez la classe main dans le champ "Main class" (com.neurotec.samples.AbisApplication) ;
- cliquez sur le bouton "Apply" et "Debug ".

c. Les variables d'environnement

- Cliquez sur le menu démarrer de Windows ;
- cliquez sur "Paramètres système avancés" ;
- cliquez sur "Variables d'environnement" ;
- dans "Variables Système" cliquez sur le bouton "Nouveau" ;
- saisissez JAVA_HOME dans le champ "Nom de la variable" et dans le champ "Valeur de la variable" saisissez C:\Program Files\Java\jdk1.7.0_45 ;
- cliquez sur "OK" ;
- ajoutez Apache Maven ;
- cliquez sur "Nouveau" ;
- saisissez M2_HOME dans le champ "Nom de la variable" et dans le champ "Valeur de la variable" saisissez C:\Program Files (x86)\apache-maven-3.1.1\bin ;
- ajoutez les deux variables au Path.

II. Configuration/Module pour Android

1. Configuration de l'IDE

- Téléchargez la version pour Windows, et décompressez le fichier zip sur le c:\ ;
- ajoutez le chemin du SDK Android à la variable Path du système ;
- lancez Eclipse, puis allez dans le menu **Help** > **Install New Software**, puis cliquez sur le bouton ***Add*** et remplissez le formulaire ;
- dans le champ ***Name*** écrivez par exemple *Plugin Android* et dans le champ ***Location*** indiquez l'adresse suivante : *https://dl-ssl.google.com/android/eclipse/* puis cliquez sur ***OK ;***
- cochez ***Developer Tools*** puis cliquez sur ***Next*** et encore une fois sur ***Next*** ;
- cochez « *I accept the terms of the license agreements* » puis cliquez sur ***Finish***;
- une fois l'installation finie redémarrez Eclipse. Une fois Eclipse relancé, allez dans le menu ***Windows*** > ***Préférences*** > ***Android*** puis ***SDK Location*** allez chercher le dossier dans lequel se trouve le SDK décompressé puis cliquez sur ***OK***.

2. Mise en place du web service

Etape1 : Téléchargement des outils

- Téléchargez Axis2 (la distribution binaire) ;
- décompressez Axis sur le c:\ ;
- téléchargez Apache Tomcat ;
- décompressez Apache Tomcat sur le c:\ ;
- ajoutez le chemin de Apache Tomcat à la variable Path du système.

Etape2 : Configurations d'Eclipse

- Lancez Eclipse ;
- dans Eclipse allez dans Windows -> Préférences et sélectionnez serveur -> Runtime Environnements ;
- cliquez sur add et indiquez le chemin de Apache Tomcat ;
- cliquez sur terminez ;
- ensuite, cliquez sur service Web -> Préférences Axis2 et indiquez le chemin de Apache Axis2 distribution binaire ;
- créez le projet qui devra contenir les méthodes du web service ;

- faites un clic droit sur la clase contenant les méthodes web service->create webservice ;
- cliquez sur suivant et choisissez les différentes méthodes publiques ;
- cliquez sur finish ;
- axis génère automatiquement le web service ;
- copiez les fichiers jar et placez-les dans webcontent->web inf->lib situé dans le web service généré ;
- faites un clic droit sur le nom du web service export->war file ;
- dans destination, indiquez le dossier webapps se trouvant dans Apache Tomcat ;
- cliquez sur Finish.

3. Procédure de déploiement/Module pour PC

Pour le déploiement de l'application de l'application nous avons utilisé Apache Tomcat. Apache Tomcat est un serveur d'application développé conçu pour le déploiement d'application.

Le système étant développé avec Java SE, le problème de déploiement s'est vite posé. Il nous a fallu utilisés les applets pour pouvoir accéder à notre application à travers le serveur d'application. Pour le déploiement les étapes suivantes sont nécessaires :

- télécharger Apache Tomcat ;
- placez Apache Tomcat sous c:/ ;
- générer le dossier de déploiement :
    - lancer cmd.exe ;
    - se placer sur le projet ;
    - taper mvn clean package;
    - taper mvn package –Papplet.
- placer le dossier sous apache-tomcat/webapps ;
- lancer Apache Tomcat à travers l'url suivante http://localhost:8080/;
- cliquez sur le menu Tomcat Manager et choisissez le projet pour le lancer sur le navigateur.

Le déploiement de l'application sous Apache-Tomcat nous permet d'accéder à l'application sous le navigateur à travers une url. Cela nous permet également d'effectuer une mise à jour et une maintenance facile de l'application.

III. Présentation de quelques écrans

1. Ecrans du Module pour PC

- *Ecran de connexion*

Cet écran permet à l'utilisateur d'entrer son mot de passe et son login afin d'accéder à la page d'accueil de l'application.

Figure VI.1 : Ecran de connexion/application sur PC

- *Page d'accueil*

La page d'accueil présente quatre menus à savoir le menu :

- Fichier : ce menu permet de se connecter de l'application et de changer de mot de passe ;
- Gestions des empreintes: ce menu comporte les sous menus *enrôler empreinte*, *vérifier empreinte* et *identifier empreinte* ;
- Configurations : Ce menu permet de faire les différentes configurations concernant le traitement des empreintes digitales .Il permet également de visualiser les enregistrements et d'y effectuer des recherches ;
- Administration : Ce menu permet la gestion des utilisateurs.

Figure VI.2 : Page d'accueil application sur PC

- *Page Enrôlement*

Pour accéder à la page *Enrôlement*, il faudrait choisir le sous menu *Enrôler* contenu dans le menu *Gestions des empreintes*. L'enrôlement d'un individu comporte plusieurs phases et chaque phase est illustrée par un écran.

- Ecran 1 : Saisie des informations de la personne à enrôler

L'écran 1 permet la saisie des informations relatives à l'identité de l'individu. Il permet également de prendre une photo directement via la webcam ou de l'attaché en tant que fichier joint.

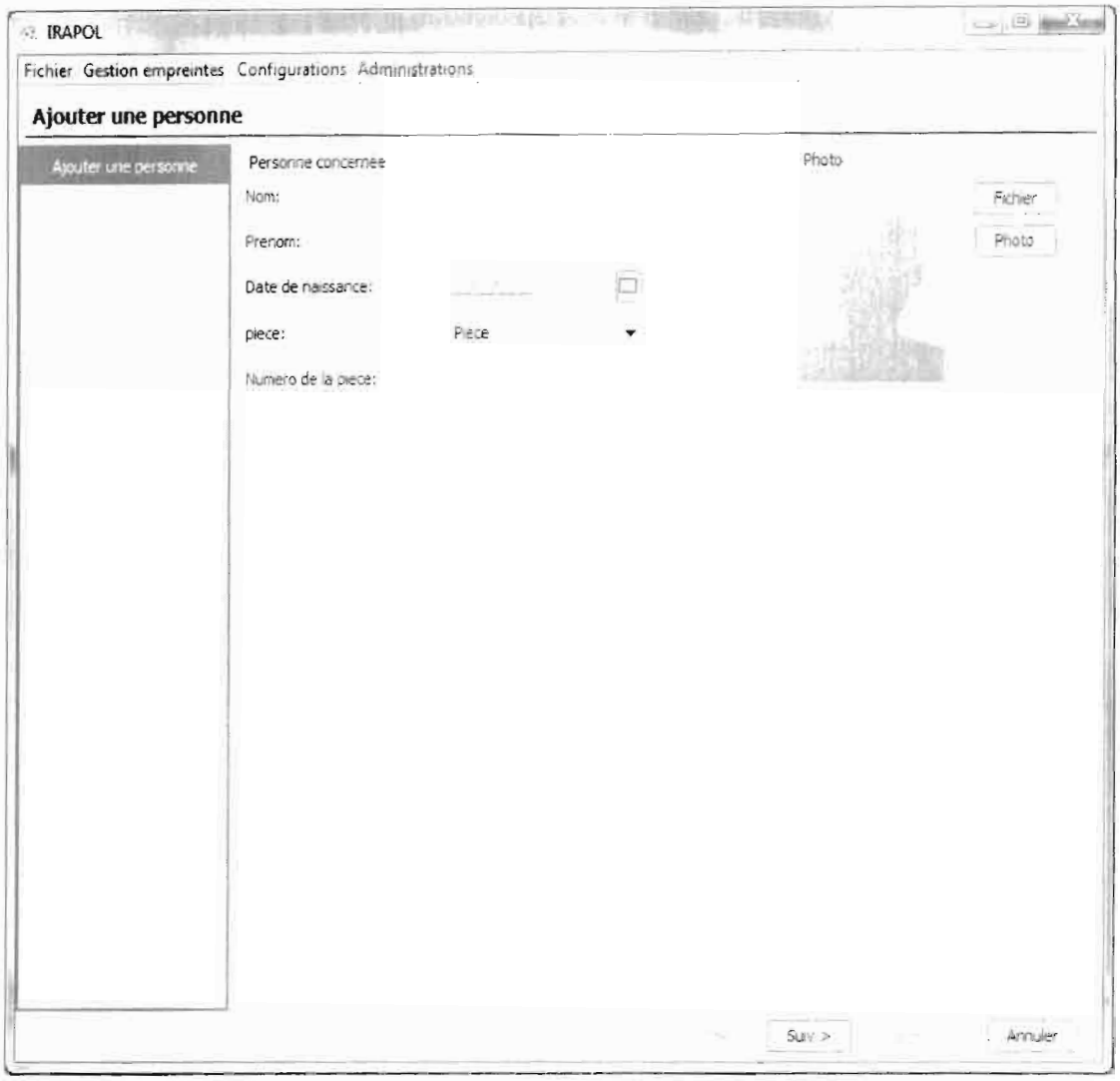

Figure VI.3 : Ecran ajout personne

- Ecran 2 : Joindre les empreintes digitales de la personne concernée

L'écran 2 permet de joindre les empreintes digitales de l'individu soit à travers un scanner, soit en tant qu'image déjà stocké sur la machine. L'écran permet également de préciser le scenario (un seul doigt avec la position, les dix doigts sans la position, les doigts de la main gauche).

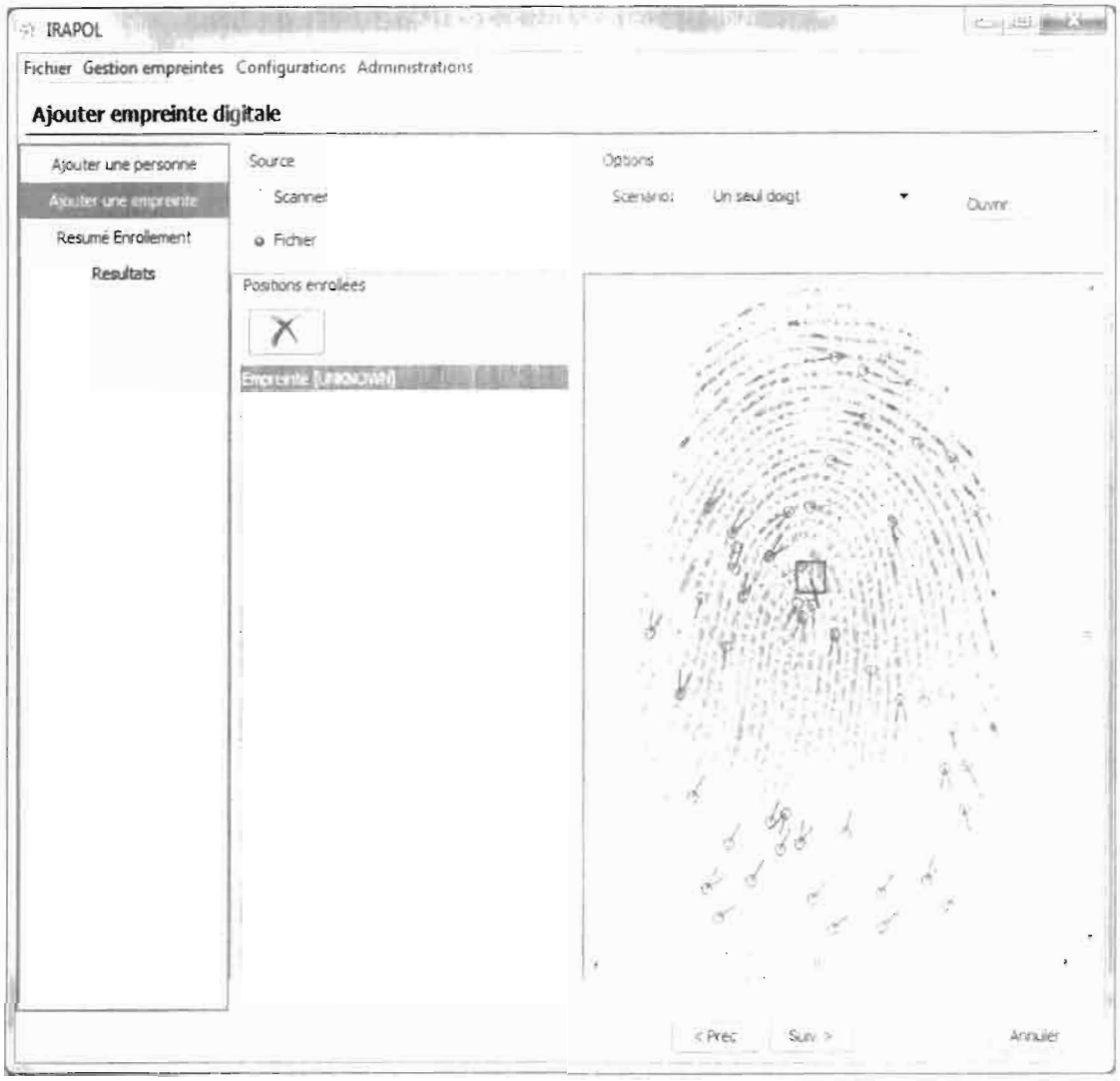

Figure VI.4 : Ecran ajout d'empreintes

- *Page Identification*

  - Ecran 1 : Ajouter l'empreinte de la personne à identifier

Pour identifier un individu, l'utilisateur doit accéder au sous menu *Identifier empreinte*. L'écran d'ajout d'une empreinte permet d'ajouter les différentes empreintes à identifier soit par le scanner soit sous forme de fichier image.

Figure VI.5 : Ecran empreinte à identifier

- Ecran 2 : Résultat de l'identification

Une fois les différentes empreintes renseignées au niveau de la page d'ajout des empreintes. l'utilisateur peut cliquer sur le bouton suivant afin de voir le résultat de l'identification. Si une des empreintes est identifiée, les informations de l'individu ainsi que sa photo sont affichées sur la page de résultat, dans le cas contraire un message s'affiche pour indiquer qu'aucune des empreintes n'existe dans la base de données.

Figure VI.6 : Ecran résultat identification

2. Ecrans du module pour Android

- *Page de connexion*

La page de connexion permet à l'utilisateur de pouvoir accéder aux différents menus de l'application. L'utilisateur doit entrer son login et son mot de passe afin d'y avoir accès.

Figure VI.7: Ecran de connexion/Application Android

- *Page d'accueil*

La page d'accueil présente quatre menus à savoir le menu :

- Chargement : ce menu permet de charger les empreintes digitales pour l'enrôlement, l'identification ou la vérification;
- Enrôler : ce menu permet de saisir les informations de l'individu et d'effectuer l'enregistrement dans la base de données ;
- Identifier : ce menu permet d'effectuer une identification à partir de l'empreinte chargée ;
- Vérifier : ce menu permet d'effectuer une vérification à partir de l'empreinte chargée.

Figure VI.8 : Ecran d'accueil/Application Android

- *Page d'Enrôlement*

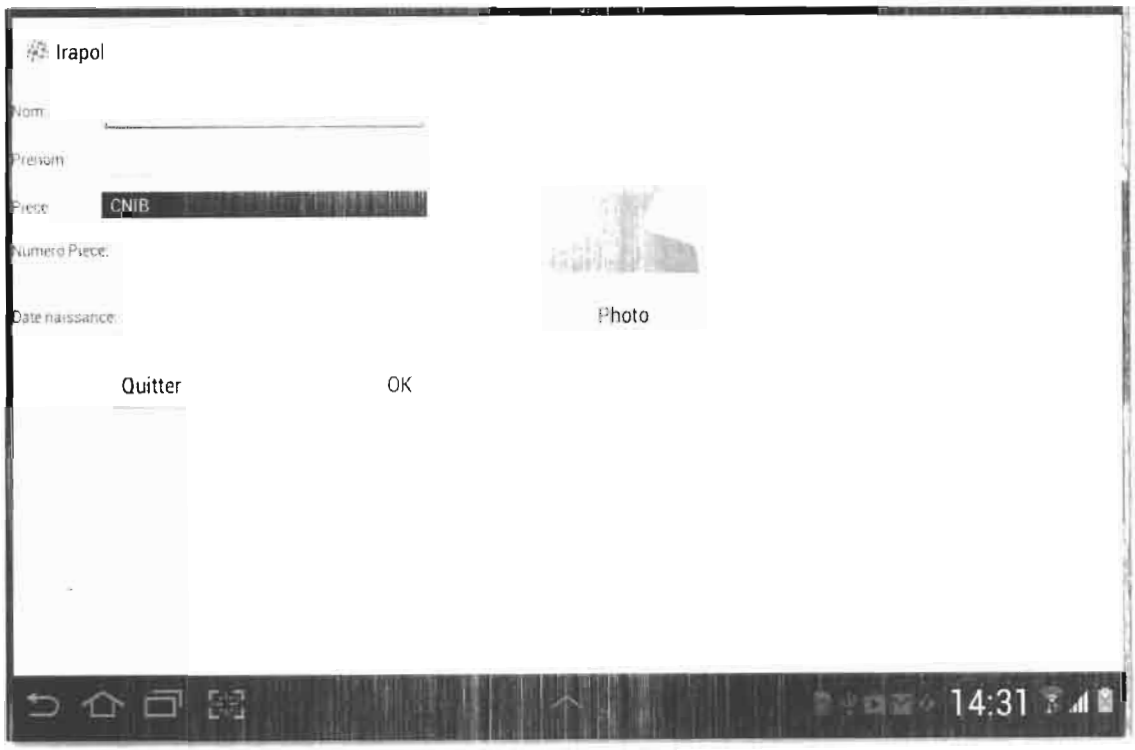

Figure VI.9 : Ecran enrôlement/Application Android

Conclusion

Dans ce chapitre nous avons pu présenter les configurations nécessaires quant à l'implémentation des deux modules. Les étapes de déploiement avec le serveur d'application Apache-Tomcat ont également été présentées ainsi que quelques écrans de l'application développée aussi bien pour PC que pour Android conformément aux besoins de l'utilisateur.

Conclusion générale

Dans ce document nous avons rapporté le travail effectué au cours de notre stage au sein de la Société SITI-INFO. Ce travail avait pour but la conception et la réalisation d'un système biométrique d'identification par empreinte digitale.

Nous avions ainsi donc pour objectif la conception d'un système biométrique fonctionnel aussi bien sur PC que sur Android qui puisse apporter une solution quant aux problèmes rencontrés par la police et de la gendarmerie nationale à savoir l'impossibilité d'identification avec uniquement les empreintes digitales, l'impossibilité d'identification en tout lieu par le manque de système mobile.

Pour parvenir à nos fins, nous avons utilisé 2TUP comme processus d'analyse de notre projet, le SDK Megamatcher qui fournit des outils nécessaires quant au développement de systèmes biométriques, eclipse comme IDE, java comme langage de programmation et oracle **10 g** pour la base de données.

A la fin de notre stage, nous pouvons noter des points de satisfaction quant au travail réalisé ainsi que l'ensemble des connaissances que nous avons pu assimiler tout au long du stage. Notons que nous avons pu réaliser le Module pour PC ainsi que le Module pour Android et effectuer des tests qui se sont montrés très concluants. Le stage nous a permis de renforcer nos connaissances quant aux traitements d'image, de nous familiariser avec le développement sous Android ainsi que le développement de services web.

Vu la nécessité du projet nous jugeons nécessaires qu'on prenne en compte d'autres critères d'identification comme l'iris et la voix qui seront des modules complémentaires et permettront certainement une identification plus fiable et multicritères.

Bibliographie

Livres

[1] PASCAL ROQUES, Franck Vallée, UML2 en ACTION, Eyrolles, 2008 ;

[2] Xavier Philippeau, Les filtres usuels en traitement d'images, Develloppez.com, 2007 ;

[3] Humbert Florent, Techniques de débruitage d'images, 2008 ;

Rapports de stage

[4] BOUDJELLAL Sofiane « Détection et identification de personne par méthode biométrique »,2007 .

[5] Dusenge Tony, La Reconnaissance des Empreintes Digitales, 2009 ;

[6] Stéphane DORVAL, Biométrie appliquée à la sécurité domestique, 2008.

Webographie

[7] http://www.neurotechnology.com/

[8] http://fr.wikipedia.org/wiki/Empreinte_digitale

[9] http://biometricgroup.com

[10] http://kobjects.org/ksoap2/index.html

[11] http://fr.wikipedia.org/wiki/Service_web

[12] http://www.commentcamarche.net/contents/1244-web-services

NB : Ces liens ont été visités dans la période du 16 décembre 2013 au 16 juin 2014