



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique
Université Mouloud MAMMERY de Tizi-Ouzou
Faculté de Génie électrique et Informatique
Département Informatique

Projet de fin d'étude pour l'obtention du diplôme de Master
Option : Réseaux, Mobilité et Systèmes Embarqués

Contrôle d'accès et Gestion de la bibliothèque par les Tags RFID

Réalisé par :

Mr. SID ALI Mourad

Mr. SLIMANI KHALED

Encadrés par :

Mr. DAOUI Mehammed

Promotion : 2012-2013

Remerciements

D'abord nous remercions le bon dieu de nous avoir donné santé, courage, volonté et foi pour réaliser ce travail.

Nous tenons à exprimer notre profonde gratitude à notre encadreur Mr DAOUI Mehammed, pour tout ce qu'il nous a apporté comme aide, connaissances et conseils pour l'accomplissement de ce travail.

Nous remercions vivement les membres du jury pour avoir accepté d'évaluer notre travail.

Aux enseignants de département d'informatique de l'UMMTO pour l'effort qu'ils ont déployé afin d'assurer notre Formation, pour leurs compétences, et surtout leur modestie.

Nous aimerions aussi remercier nos familles, nos amis, ainsi que tous ceux qui ont contribué de prêt ou de loin à la réalisation de ce travail.

Sommaire

Introduction générale

Les nouvelles technologies de l'information et de la communication ont un rôle fondamental dans notre société moderne, en raison de leurs effets positifs sur les plans économiques et sociaux.

L'introduction de ces technologies de l'information participe ainsi à la croissance économique en facilitant l'échange et la diffusion des informations. Les technologies d'identification font partie de ces technologies. Elles trouvent leurs applications dans des domaines très divers tels que la distribution, la logistique, la traçabilité, la sécurité et les loisirs.

Les technologies d'identifications étaient soit passives : marquage, code barre... soit nécessitant un contact : carte bancaire, carte d'appels téléphoniques...

Grâce au développement récent des systèmes sans fil et de la micro-électronique, une nouvelle technologie d'identification sans contacts a vu le jour : la technologie d'identification par les radios fréquences (ou RFID pour Radio-Frequency Identification). Cette technologie a la particularité de fonctionner à distance, sur le principe suivant : un lecteur émet un signal radio et reçoit en retour les réponses des étiquettes (ou tags) qui se trouvent dans son champ d'action.

La technologie RFID est utilisée largement dans le contrôle d'accès, la traçabilité, l'authentification d'objet, l'étiquetage, la logistique, etc.

Les récents progrès technologiques dans les systèmes embarqués ont permis l'apparition de la technologie NFC (Near field Communication), qui est une technologie sans contact au même titre que la technologie RFID, elle est destinée à être embarquée dans tout système électronique communiquant voulant bénéficier de la possibilité d'échanger de manière sécurisée et simple des informations. Parmi ces objets, nous pouvons citer: les Smartphones, PDAs et tablettes.

Le but de notre travail est d'étudier les technologies RFID et NFC, à travers une application permettant le contrôle d'accès et la gestion de l'emprunt et la restitution d'ouvrage dans une bibliothèque. Notre application est implémentée sur un Smartphone Android disposant d'un lecteur NFC.

Notre mémoire est structuré en quatre chapitres : Le premier est consacré à la description des systèmes embarqués en général. Dans le second chapitre, nous présenterons la

technologie RFID et NFC : principes physiques et protocoles. Le troisième chapitre est consacré à l'étude détaillée de notre projet. Enfin, le dernier chapitre est consacré à la phase d'implémentation et tests de notre application.

Introduction :

Si nous observons l'environnement de notre société, nous pouvons constater qu'une multitude de systèmes embarqués nous entoure, à partir des téléphones portables et jusqu'aux systèmes de contrôle des voitures ou des avions. Ces systèmes existent dans l'industrie, mais on les trouve également très répandus dans la vie quotidienne. Dans ce premier chapitre, nous allons présenter les systèmes embarqués en se basant sur les systèmes mobiles.

I.1 Définition d'un système embarqué:

Un système embarqué est un système électronique et informatique autonome ne possédant pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur. Il est piloté dans la majorité des cas par un logiciel, qui est complètement intégré au système qu'il contrôle. Les concepteurs des systèmes électroniques sont aujourd'hui confrontés à la complexité croissante des algorithmes mis en œuvre et à la variété des cibles potentielles (FPGAs, ASIC et ASSP, etc...). Actuellement, il n'est pas rare que ces systèmes intègrent plusieurs douzaines voire des centaines de processeurs. A l'origine, ce sont des systèmes matériels et logiciels intégrés dans des avions militaires ou des missiles. Ensuite dans le civil: avions, voitures, machine à laver...! Le matériel et le logiciel sont intimement liés, ils ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC.

Mentionnons les contraintes physiques fortes: dimensions, poids, taille, autonomie, consommation, fiabilité, contraintes temporelles (temps réels).

I.2 Historique :

Nous pouvons échelonner l'évolution des systèmes embarqués comme suit :

- 1967 : Le premier système moderne embarqué reconnaissable a été le Apollo Guidance Computer Environ, un millier de circuits intégrés identiques (portes NAND).
- 1960-1970 : Missile Minute man, guidé par des circuits intégrés.
- 1971 : Intel produit le 4004, Ce premier microprocesseur, était le premier circuit intégré incorporant tous les éléments d'un ordinateur dans un seul boîtier: unité de calcul, mémoire, contrôle des entrées / sorties. Alors qu'il

fallait auparavant plusieurs circuits intégrés différents, chacun dédié à une tâche particulière, avec ce type de microprocesseur un seul pouvait assurer autant de travaux différents que possible. Très rapidement, des objets quotidiens tels que fours à micro-ondes, télévisions et automobiles à moteur à injection électronique ne tardèrent pas à être équipés de microprocesseurs.

- 1972 : lancement de l'Intel 8008, premier microprocesseur 8 bits (48 instructions, 800kHz).
- 1974 : lancement du 8080, premier microprocesseur largement diffusé. 8 bits, (64KB d'espace adressable, 2MHz - 3MHz).
- 1978 : création du Z80, processeur 8 bits.
- 1979 : création du MC68000, processeur 16/32 bits.

I.3 Composition d'un système embarqué :

- Une partie matérielle caractérisée par la performance.
 - Micro-processeur, contrôleurs, coprocesseurs, DSP.
 - Mémoires.
 - ASIC.
 - Interfaces d'entrées/sorties
- Une partie logicielle caractérisée par la flexibilité

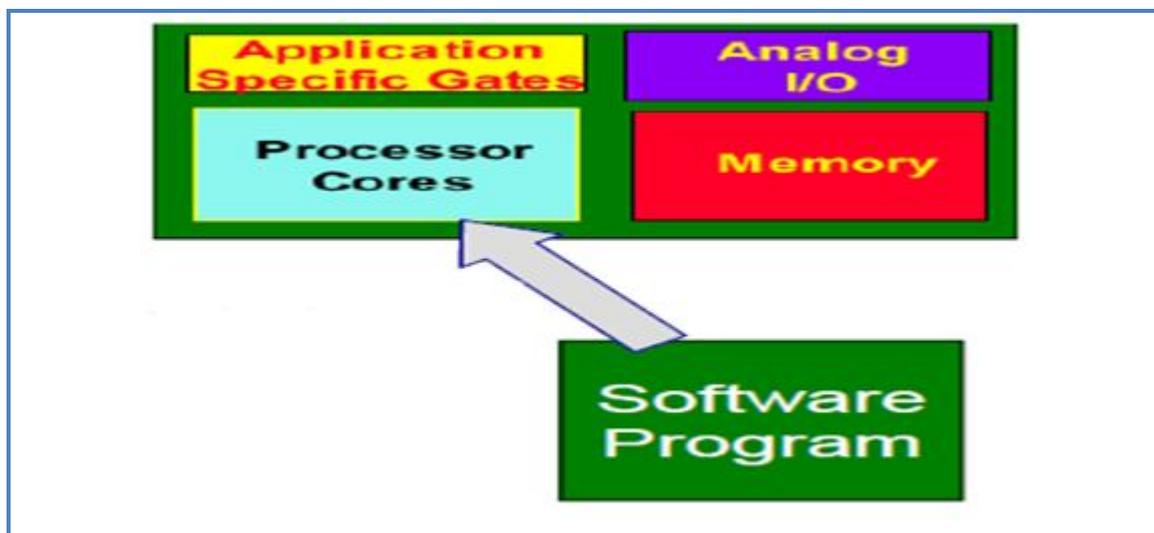


Figure I.1: Composition d'un système embarqué. [1]

I.4 Caractéristiques des systèmes embarqués :

- **Fonctionnement en Temps Réel :**
 - Réactivité : des opérations de calcul doivent être faites en réponse à un événement extérieur (interruption matérielle).

- La validité d'un résultat (et sa pertinence) dépend du moment où il est délivré.
- Rater une échéance va causer une erreur de fonctionnement.
 - Temps Réel dur: erreur sévère.
 - Temps Réel mou: dégradation non dramatique des performances du système.
- Beaucoup de systèmes sont « multirate »: ils permettent le traitement d'informations à différents rythmes.
- **Faible encombrement, faible poids :**
 - Electronique « *pocket PC* », applications portables où l'on doit minimiser la consommation électrique (bioinstrumentation...).
 - Difficulté pour réaliser le packaging afin de faire cohabiter sur une faible surface électronique analogique, électronique numérique, RF sans interférences.
- **Faible consommation :**
 - Batterie de 8 heures et plus (PC portable : 2 heures).
- **Environnement :**
 - Température, vibrations, chocs, variations d'alimentation, interférences RF, corrosion, eau, feu, radiations.
 - Le système n'évolue pas dans un environnement contrôlé.
 - Prise en compte des évolutions des caractéristiques des composants en fonction de la température, des radiations...
- **Fonctionnement critique pour la sécurité des personnes. Sûreté :**
 - Le système doit toujours fonctionner correctement.
 - Sûreté à faible coût avec une redondance minimale.
 - Sûreté de fonctionnement du logiciel
 - Système opérationnel même quand un composant électronique lâche.
 - Choix entre un design tout électronique ou électromécanique.
- **Beaucoup de systèmes embarqués sont fabriqués en grande série et doivent avoir des prix de revient extrêmement faibles, ce qui induit :**
 - Une faible capacité mémoire.
 - Un petit processeur jusqu'à (4 bits). Petit mais en grand nombre.
- La consommation est un point critique pour les systèmes avec autonomie.

- Une consommation excessive augmente le prix de revient du système embarqué car il faut alors des batteries de forte capacité.
- **Faible coût :**
 - Optimisation du prix de revient.

I.5 Domaine d'application:

Les domaines dans lesquels on trouve des systèmes embarqués sont de plus en plus nombreux :

- Transport : Automobile, Aéronautique (avionique), etc.
- Astronautique : fusée, satellite artificiel, sonde spatiale, etc.
- Militaire : missile.
- Télécommunication : Set-top box, téléphonie, routeur, pare-feu, serveur de temps, téléphone portable, etc.
- Électroménager : télévision, four à micro-ondes.
- Impression : imprimante multifonctions, photocopieur, etc.
- Informatique : disque dur, Lecteur de disquette, etc.
- Multimédia : console de jeux vidéo, assistant personnel.
- Guichet automatique bancaire (GAB).
- Équipement médical.
- Automate programmable industriel.
- Métrologie.

I.6. Systèmes d'Exploitation Embarqués:

On a vu auparavant que les caractéristiques des systèmes embarqués sont différentes de ceux des ordinateurs traditionnels. Ils sont caractérisés par une faible puissance de calcul, un faible espace de stockage, une autonomie limitée. Ceci rend les systèmes d'exploitations traditionnels (comme Windows) inutilisable et inadapté. Des systèmes d'exploitations pour mobile ont été conçus et se concentrent entre autres sur la gestion de la connectivité sans fil et celle des différents types d'interface, les plus connus sont actuellement Android et IOS, dans cette partie nous présentons brièvement les différents systèmes d'exploitations existants pour mobile.

I.6.1. Microsoft Windows Mobile: [2]

Windows Mobile est le nom générique donné à différentes versions de Microsoft Windows conçues pour des appareils mobiles tels que les Smartphones ou les Pockets PC. Ces systèmes d'exploitation permettent à des logiciels Microsoft tels que Microsoft Office ou Windows Live Messenger de fonctionner sur un téléphone ou un ordinateur portable. L'avantage majeur est de pouvoir recevoir des courriels en temps réel. En 2008, Microsoft a vendu 20 millions de licences Windows Mobile aux constructeurs d'assistant personnel (PDA). Cependant depuis Windows Phone 8, Microsoft propose des fonctions avancées pour les entreprises en offrant, par exemple, un espace d'applications réservé aux entreprises. Il a été lancé le 21 octobre 2010 en Europe et le 8 novembre 2010 aux États-Unis.

Le tableau ci-dessous donne un récapitulatif sur les différents systèmes Windows Mobile :

Version	Année de mise en œuvre
Windows CE 1 Novembre 1996	1996
Windows CE 2 Novembre	1987
Windows CE 2.1 Juillet	1998
Windows CE 3	1999
Windows CE .NET	2000
Pocket PC 2000	2001
Pocket PC 2001	2003
Windows Mobile 2000 SE (Second Edition)	2004
Windows Mobile 2003 5.0 (Pocket PC 2003)	2005
Windows Mobile 6.0	2006
Windows Mobile 6.1	2008
Windows Mobile 6.5	2008
Windows Phone 7	2010

Tableau I.1: Evolution des versions Microsoft Windows Mobile.

Nous ferons dans ce qui suit une description de quelques-uns des principaux systèmes Windows Mobile :

- A. **Windows CE :** Windows CE (parfois abrégé Win CE) est une variation de Windows pour les systèmes embarqués et autres systèmes minimalistes, utilisé notamment dans les PC de poche. Il utilise un noyau distinct des autres Windows plutôt qu'une version allégée et supporte les architectures processeur Intel x86 et similaires, MIPS (jusqu'à CE

3.0), ARM et aussi Hitachi SH. Windows CE est optimisé pour les appareils possédant une faible capacité de stockage, le noyau peut tourner avec moins de 1 Mo de mémoire vive. Les systèmes sont souvent produits sans disque de stockage et peuvent être pensés pour ne pas pouvoir être étendus (par exemple, le système peut être écrit en ROM). Windows CE est conforme à la définition d'un système d'exploitation temps réel. Il supporte 256 niveaux de priorité et permet l'inversement de priorité. À l'inverse des systèmes ressemblant à UNIX, l'unité fondamentale d'exécution est le thread permettant d'obtenir des programmes plus simples et plus rapides.

- B. **Pocket pc** : Microsoft Pocket PC est l'évolution des versions Windows CE. Il existe actuellement 3 grandes versions de Microsoft Pocket PC (2000, 2002, 2003). Ces systèmes sont conçus pour être utilisés sur des machines dites de format tablette possédant, comme principal moyen d'interface utilisateur, un écran tactile d'une résolution de 320*240 pour les Pocket PC 2000, 2002, 2003, et de 640*480 pour certains Pocket PC 2003 Seconde Edition et 5.0 (Toshiba e830, Asus A730, Dell Axim x50v). Ils possèdent en standard un certain nombre d'applications préinstallées tels que Pocket Word, Pocket Excel, Pocket Outlook, Pocket Internet Explorer. Les applications Pocket PC sont incompatibles avec les versions de Windows pour les ordinateurs de type PC mais peuvent permettre de relire des documents transférés via l'outil de synchronisation ActiveSync. Ce système fonctionne sur des machines équipées des processeurs ARM ou XScale.
- C. **Windows Mobile 2003** : Cette version offre quelques possibilités nouvelles pour la prochaine génération d'assistants personnels et de Smartphones. Elle supporte le mode paysage, les écrans carrés (240x240 ou 480x480) et offre une fonction d'étirement des pixels qui permet de rendre compatibles avec l'affichage VGA (480x640) les applications développées en QVGA (240x320). Des améliorations graphiques pour une apparence en 3D sont disponibles. Le support Wi-Fi est en place avec la prise en charge du WPA (Wi-Fi Protected Access).

I.6.2 iOS : [3]

iOS, anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch, et l'iPad. Il est dérivé de Mac OS X dont il partage les fondations. Le système d'exploitation occupe moins d'un demi-gigaoctet (Go) de la capacité mémoire totale de l'appareil. A fin d'utiliser le kit de développement nécessaire à réaliser des

applications et les publier il faut adhérer au programme des développeurs Apple, pour la somme de 99 \$ par an.

I.6.3 Palm OS :

Palm OS est un système d'exploitation pour appareils mobiles édité par PalmSource, séparé en 2003 de la société Palm (PalmOne depuis). Initialement conçus pour les PDA de Palm, Palm OS est adopté par beaucoup d'autres fabricants à l'instar de : Handspring (séparé de Palm puis racheté par PalmOne), Sony, IBM, Qualcomm, Symbol, Tapwave, AlphaSmart, Fossil, Garmin, HandEra, Acceca, AlphaSmart, GSPda et Samsung.

Le tableau I.2 suivant résume l'évolution des systèmes Palm OS :

Version	PDA cible
Palm OS 1.0	Pilot 1000 and 5000
Palm OS 2.0	Palm Pilot Personal and Professional
Palm OS 3.0, 3.1, 3.3, 3.5	Palm III series
Palm OS 4.0	Palm500 series
Palm OS 5.0	Tungsten T
Palm OS 1.0	Garnet (5.x) Multi plate-forme
Palm OS 1.0	Cobalt (6.x) Multi plate-forme

Tableau I.2: Evolution des versions de Palm OS.

Palm OS vient d'être abandonné par la compagnie qui l'avait acheté il y a quelques années. Access a annoncé le 10 Octobre 2006 que le système Palm OS allait être remplacé définitivement par Access Linux Platform. Les applications Palm OS seront néanmoins compatibles car cette version de Linux embarquée aura une couche logicielle pour supporter les programmes Palm OS.

I.6.4 Symbian OS : [4]

Symbian OS est un système d'exploitation pour téléphones portables (on parle de « Smartphone ») et PDA conçu par Symbian ltd.

Il est né d'un consortium de différents constructeurs. Il dispose de nombreuses API spécifiques pour la communication mobile voix et données, et utilise des protocoles standard de communication réseau : IPv4/IPv6, WAP, MMS, Bluetooth, GPRS/UMTS, Java, SyncML

I.6.5 BlackBerry OS : [3]

BlackBerry OS est un système d'exploitation propriétaire pour téléphone mobile de la gamme BlackBerry, conçu par la société canadienne Research In Motion (RIM), maintenant BlackBerry.

Il s'agit d'un système multitâche. Le système est surtout connu pour son support natif des courriels à travers le protocole Mobile information device profile (MIDP 1.0), et plus récemment un sous-ensemble de MIDP 2.0 qui permet une synchronisation complète avec les messageries d'entreprise telles que Microsoft Exchange ou IBM Lotus Domino.

I.6.6 Linux OS :

Le système Linux doit sa venue sur le marché des PDA à la série Zaurus de Sharp qui a réservé ses assistants personnels au marché asiatique. Mais il faut noter qu'avant même l'annonce du Zaurus SL-5500, les passionnés bricolaient déjà leur iPaq et installaient leur propre version de Linux. C'est d'ailleurs l'existence de cette florissante communauté qui a motivé Sharp à adopter Linux. Plusieurs distributions de linux pour PDA ont été produites, à l'instar de : OpenEmbedded (extension de openzaurus), Lineo Embedix Linux, familiar (Ipaq), linupy (version lite de Gnome, distribution officielle de Yopi), Linux DA Embedded O/S™, etc.

I.6.7 Bada: [2]

Bada est le système d'exploitation pour Smartphones Samsung Wave de Samsung basé sur le système d'exploitation propriétaire SHP OS. Il est sorti début 2010, après avoir été dévoilé au Mobile World Congress de Barcelone de la même année.

Bada est, au 27 août 2011, le 3e en part de marché en France et équipe plus d'un million de Smartphones.

Le 25 février 2013, Samsung a annoncé qu'il va arrêter le développement de Bada, et se concentrer, à la place, sur le développement d'un autre système d'exploitation (Tizen).

I.6.8 Android: [5]

Android est un système d'exploitation open-source pour Smartphones, PDA et autres terminaux mobiles, conçu par Android, une start-up rachetée par Google en juillet 2005. Il existe d'autres types d'appareils possédant ce système d'exploitation tels que les téléviseurs et les tablettes.

Afin de promouvoir ce nouveau système d'exploitation ouvert, Google a su fédérer autour de lui un consortium d'une trentaine d'entreprises : l'Open Handset Alliance (OHA) créée officiellement le 5 novembre 2007. Toutes ces entreprises interviennent, plus ou moins directement, dans le marché de la téléphonie mobile. Le but de cette alliance est de mettre en place des normes ouvertes dans le domaine de la téléphonie mobile. Les développeurs d'application Android pourront accéder aux fonctionnalités du cœur du téléphone via une API très fournie.

Android aura comme principaux concurrents Apple avec l'iPhone, Microsoft et son Windows Mobile et Nokia avec Symbian mais également des solutions libres telles que LIMO ou OpenMoko.

a) Historique :



Figure I.2: Aperçu des versions d'Android.

La première version d'Android apparaît en octobre 2008 et n'avait pas de nom officiel. Cette version s'est avérée être la **Béta** du système.

La version 1.5 **Cupcake** corrigea le manque d'API et rendit le système plus utilisable.

Depuis, **Android 1.6**, **2.0** et **2.1** ont apporté d'importantes améliorations respectivement sur les fonctionnalités et sur l'interface graphique du système.

Android 2.2 Froyo a fortement mis l'accent sur la synergie avec Internet. L'envoi d'applications et de liens instantanés depuis un ordinateur est désormais possible. Aussi, Google annonce que le navigateur chrome intégré à Android 2.2 est le navigateur mobile le plus rapide au monde grâce à l'intégration du moteur JavaScript V8.

Android 3.0 Honeycomb est spécialement étudié pour les tablettes tactiles. Les premiers modèles ont été annoncés en 2011.

On y apprend quelques nouveautés comme la prise en charge de la vidéoconférence via Gtalk (Google Talk), la nouvelle interface Gmail ou encore le lecteur de livre électronique Google.

Android 4.0 IceCreamSandwich unifie le développement des interfaces Smartphone, tablette, télévision connectée et système embarqué. Avant la version 4.0, le développement d'une application pour une tablette n'était pas compatible à un Smartphone car de tailles d'écrans différentes.

Android 4.2.2 Jelly Bean Basé sur un nouveau kernel Linux 3.1.10, *Jelly Bean* est une mise à jour incrémentale avec le but premier d'améliorer l'interface utilisateur en matière de fonctionnalité et de performance. L'amélioration des performances se fait via le « *Project Butter* », qui utilise une anticipation tactile, et le passage à l'affichage de 60 images par seconde pour créer une interface utilisateur harmonieuse et fluide.

Après ce bref historique de l'évolution des versions d'Android depuis sa création, voici l'architecture générale du système basé sur un noyau Linux et nous présentons les avantages à utiliser ce dernier.

b) Architecture du système d'exploitation Android :

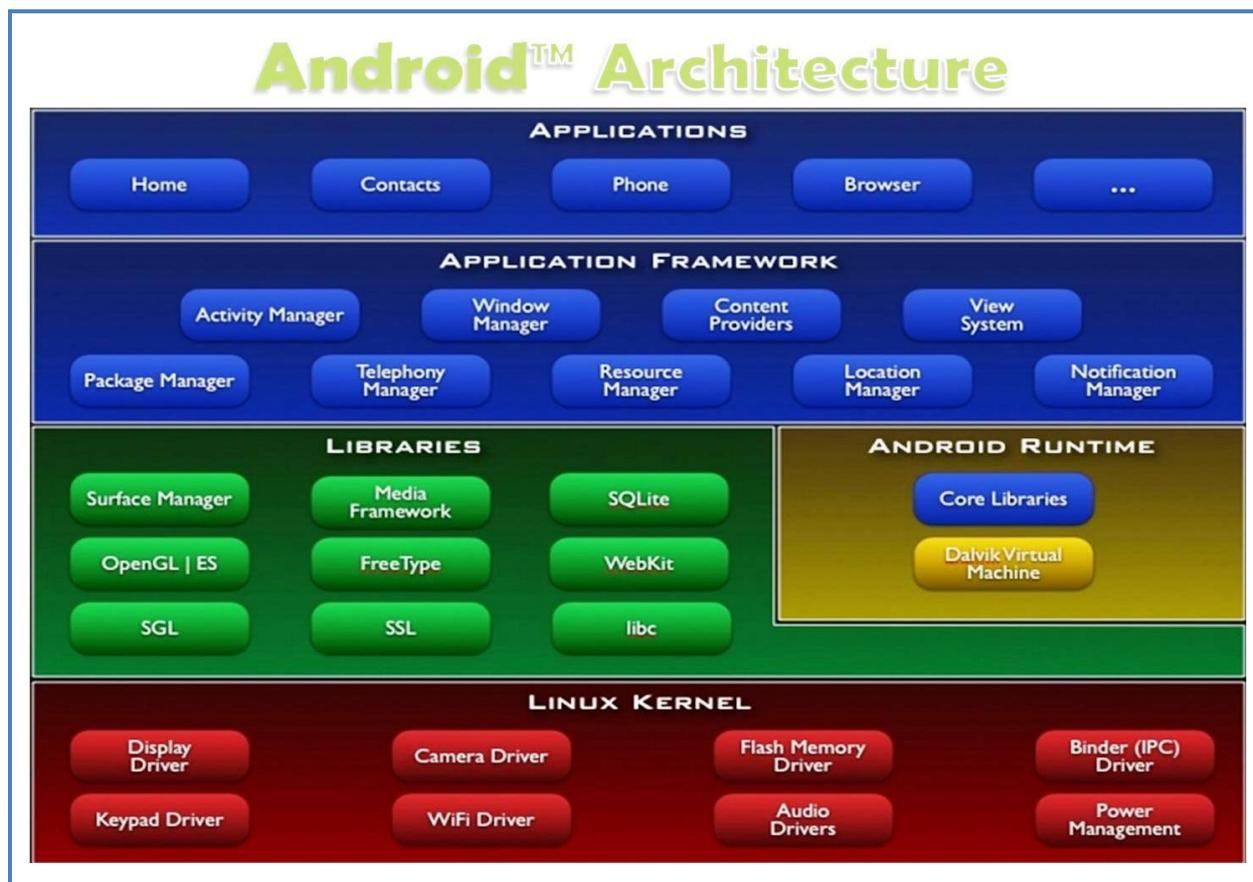


Figure I.3: Architecture d'Android.

Android est basé sur un kernel (noyau) linux. Au-dessus de cette couche, on retrouve les bibliothèques C/C++ utilisées par un certain nombre de composants du système Android.

Au-dessus des bibliothèques, on retrouve l'Android Runtime. Cette couche contient les bibliothèques cœurs du Framework ainsi que la machine virtuelle exécutant les applications.

Au-dessus de la couche "Android Runtime" et des bibliothèques cœurs, on retrouve le Framework permettant au développeur de créer des applications. Enfin au-dessus du Framework, il y a les applications.

- **Applications :**

Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur... toutes écrites en JAVA.

- **Framework de développement :**

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs sont libres de profiter du matériel périphérique et informations sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état, etc.

Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris:

- Un jeu extensible de vues qui peuvent être utilisées pour construire une application.
- Des fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications (telles que les Contacts), ou de partager leurs propres données.
- Un gestionnaire de ressources.
- Un gestionnaire de notification qui permet à toutes les demandes d'afficher des alertes personnalisées dans la barre d'état.
- Un gestionnaire d'activité qui gère le cycle de vie des applications et propose une navigation commune.

- **Android Runtime :**

Android inclut un ensemble de bibliothèques de base offrant la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java.

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour que le dispositif puisse faire tourner

plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik (. DEX), un format optimisé pour ne pas encombrer la mémoire.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

- **Bibliothèques :**

Android dispose d'un ensemble de bibliothèques C / C++ utilisées par les différents composants du système. Elles sont offertes aux développeurs à travers le Framework Android.

- **Linux Kernel :**

Android est basé sur un kernel linux 2.6 (3.1 sur android 4.2.2) mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X window system).

Un kernel avec différents patches est utilisé pour la gestion de l'alimentation, le partage mémoire, etc. permettant une meilleure gestion de ces caractéristiques pour les appareils mobiles.

L'avantage de ce dernier est son système de gestion mémoire et de processus reconnu pour sa stabilité et ses performances. Le model de sécurité utilisé par linux, basé sur un système de permission, est connu pour être robuste et performant. Il n'a pas changé depuis les années 70.

Parmi les points qui ont poussé l'équipe en charge du noyau à décidé d'utiliser un kernel linux :

- Il fournit un système de driver permettant une abstraction avec le matériel. Il permet également le partage de bibliothèques entre différents processus, le chargement et le déchargement de modules à chaud.
- Il est entièrement open source et il y a une communauté de développeurs qui l'améliorèrent et rajoutent des drivers.

c) **Applications Android :**

Les applications Android sont constituées d'un ensemble d'activités qui permettent l'interaction avec l'utilisateur, et d'un ensemble de services qui fonctionnent en arrière plan (cachés) et s'occupent d'effectuer certains traitements.

Une activité est la composante principale pour une application Android. Elle représente l'implémentation et les interactions des interfaces.



Figure I.4: Exemple d'une Activité d'une application Android.

Une activité possède un cycle de vie bien précis, qui permet de contrôler aisément l'application de son lancement jusqu'à sa fermeture. L'application passe par plusieurs états, lors du développement on associe un comportement à chaque état, ou bien lors d'un changement d'état.

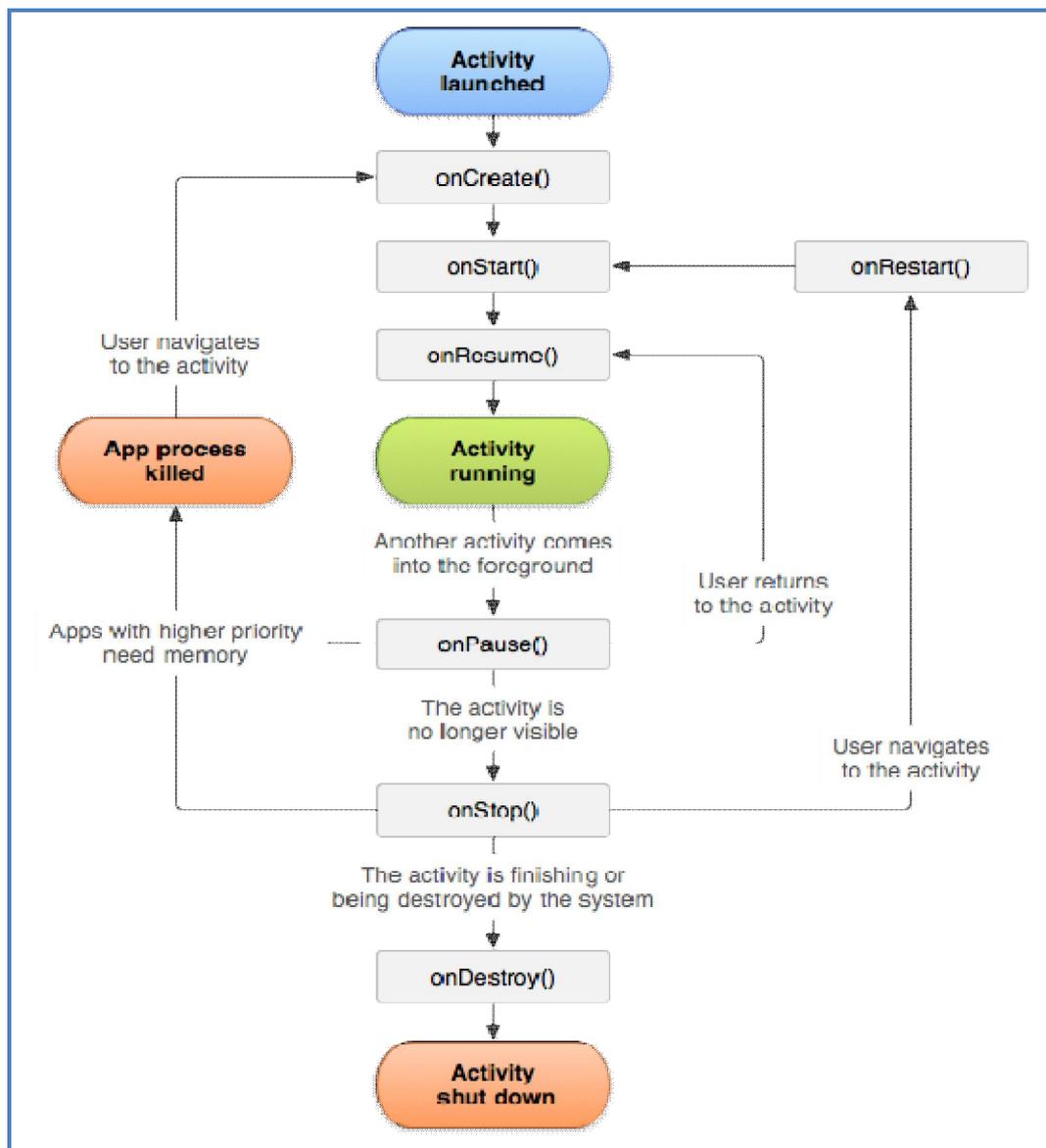


Figure I.5: Cycle de vie d'une application Android.[6]

Chaque état peut contenir une action précise à effectuer, une description brève des fonctions est présentée ci-dessous.

- `onCreate()` : Exécuté quand l'utilisateur clique sur l'icône de l'application pour une première fois
- `onRestart()`: Exécuté lorsque l'activité arrêté (mise en arrière plan) redémarre (repassé en premier plan);
- `onStart()` : Exécuté après chaque `onCreate()` ou `onRestart()`;
- `onResume()` : Exécuté à chaque passage en premier plan de l'activité;

- `onPause()` : Exécuté avant chaque arrêt de l'activité, ou avant qu'une autre application ne soit lancée (mettant l'activité actuelle en arrière plan);
- `onStop()` : Exécuté avant chaque mise en veille ou fermeture de l'activité, les ressources sont libérées.
- `onDestroy()` : Exécuté lors du de l'arrêt complet de l'activité, les ressources sont libérées (fichiers temporaires).

Android offre des avantages non négligeables, que ce soit pour les développeurs, ou pour les utilisateurs qui bénéficient grâce aux dernières versions de nombreuses fonctionnalités et d'un confort d'utilisation de qualité.

Android est :

- **Open source**

Le contrat de licence pour Android respecte les principes de l'open source, c'est-à-dire que c'est possible de télécharger à tout moment les sources et les modifier. Android utilise des bibliothèques open source puissantes, comme par exemple SQLite pour les bases de données et OpenGL pour la gestion d'images 2D et 3D.

- **Gratuit (ou presque)**

Android est gratuit, autant pour les utilisateurs que pour les constructeurs. La publication d'applications sur le Google Play, coûte la modique somme de 25\$. Ces 25\$ permettent de publier des applications à l'infini, contrairement à iPhone où il faut verser une somme de 100\$ par an.

- **Facile à développer**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire qu'il est possible d'envoyer un SMS en seulement deux lignes de code.

- **Facile à vendre**

Le Play Store (anciennement Android Market) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile.

- **Flexible**

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les Smartphones, les tablettes, la présence ou l'absence de clavier ou de trackball, différents processeurs... On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android.

Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal (si une application nécessite d'utiliser le Bluetooth, seuls les terminaux équipés de Bluetooth pourront la voir sur le Google Play).

- **Ingénieux**

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est la possibilité de combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant.

En 2013, Android s'arrogeait 75% de part de marché dans la vente de Smartphone. Apple restait en retrait avec 17,3% et Microsoft a décroché la troisième place devant Blackberry.

Conclusion:

L'évolution des technologies des systèmes embarqués offre aujourd'hui des perspectives d'intégration importante. Désormais, des systèmes complexes peuvent être intégrés dans des processeurs sur le même circuit. D'autre part, les applications de compression vidéo sous contrainte d'exécution temps réel présentent des complexités énormes. C'est pourquoi, les architectures multiprocesseurs sur puce implémentant ces applications doivent être capables de supporter plusieurs fonctionnalités et de les mettre en œuvre dynamiquement. Dans ce contexte, les méthodologies de conception doivent évoluer pour faciliter le développement, la validation et le prototypage de ces systèmes complexes. Le système et son application doivent permettre de prendre en considération les tâches répétitives et les données intensives régulières. Dans le chapitre suivant nous présentons des technologies qui utilisent ce genre de systèmes embarqués: RFID et NFC.

Introduction :

La **RFID** (**R**adio **F**requency **I**dentification) est une technologie d'identification qui utilise les radiofréquences, dans le but de tracer des objets ou des êtres vivants. Elle se range dans les systèmes d'identification automatique, au même titre que le code à barres, la reconnaissance de caractères, ou la reconnaissance de formes.

Après un bref aperçu historique de la technologie **RFID**, nous présentons d'une manière sommaire cette technologie, sa signification, son principe de fonctionnement, les éléments qu'ils la composent, tels que lecteurs et étiquettes, puis nous donnons les différentes fréquences utilisées dans un système **RFID** et quelques applications de cette technologie. Enfin, nous terminons ce chapitre par la présentation de la technologie **NFC** (extension de la **RFID**) que nous avons utilisée dans notre application.

II.1 Historique :

Nous pouvons échelonner l'évolution de la technologie RFID comme suit [7] :

- Vers les années 40 : La technologie du système RFID avait été utilisée pour la reconnaissance des avions. Les radars étaient le système de lecture qui avait pour rôle d'envoyer un signal questionnant les étiquettes des transpondeurs placés sur les avions afin de distinguer les alliés des ennemis. Et on peut considérer que le système IFF (Identity : Friend or Foe) fût la première forme d'utilisation de la technologie RFID.
- Au cours des années, la technologie a pris de l'ampleur, et en particulier vers les années 70 où elle fut une technologie protégée et à l'usage principalement militaire. Les états développés utilisaient cette technologie pour la sécurité et la protection de sites stratégiques et sensibles (secteurs nucléaires et armements lourds). Cependant vers la fin des années 70 cette technologie fut utilisée par le secteur privé et la première application commerciale fut la traçabilité (identification) du bétail dans le continent européen.
- Durant les années 80, l'évolution technologique avec l'invention des microsystèmes utilisant les circuits intégrés conduisit à l'usage des étiquettes passives. Et on remarquera à cette époque une grande diversité de types des étiquettes.
- Le début des années 90 fut l'époque de la problématique de la standardisation et de la normalisation des équipements de systèmes de la technologie RFID (étiquettes et lecteurs).

- 2003 : EAN International, Auto-ID Center, UCC (Uniform Code Council) et des industriels créent le standard EPC (Electronic Product code) intégrant les technologies RFID et Internet pour mettre en place le réseau de traçabilité des objets. [8]

Depuis lors, la technologie RFID est utilisée dans plusieurs domaines d'applications, en passant par l'identification, la traçabilité et l'analyse de données.

II.2 Définition de la technologie RFID: [9]

La RFID est une technologie faisant partie de la technologie AIDC (Automatic Identification Data Capture) utilisée pour reconnaître et identifier à plus ou moins grande distance un objet, une forme ou un individu porteur d'une étiquette capable d'émettre de données transformées en ondes radio.

La technologie de radiofréquences est généralement utilisée pour transmettre et recevoir des informations sans fils. Une grande variété d'appareils électroniques comme la télévision, la radio, le téléphone cellulaire utilise ce principe pour transmettre ou recevoir des informations. Dans la technologie RFID, l'émetteur envoie en continu un rayonnement de radiofréquences fournissant à la fois de l'énergie à l'étiquette (passive) sous forme électromagnétique et aussi l'information à transmettre. L'étiquette réagit alors et renvoie grâce à l'énergie qu'elle contient les informations vers l'émetteur.

A la différence de la technologie de code à barres qui se lit avec un laser optique, le lecteur RFID balaye ou interroge une étiquette en utilisant des signaux de radiofréquences.

II.3 Le principe de fonctionnement des systèmes RFID: [8]

Cette technologie est basée sur l'émission d'un champ électromagnétique par un « lecteur » réceptionné par une antenne couplée à une puce électronique (étiquette, transpondeur ou tag). Ce champ a pour rôle de véhiculer l'information (entre la puce et son lecteur) et l'énergie nécessaire à l'activation de la puce. Lorsque les tags sont "réveillés" par le lecteur, un dialogue s'établit selon un protocole de communication prédéfini et les données sont échangées. Le lecteur est relié à une application hôte qui récupère les informations récoltées par ce lecteur pour les traiter. Donc une application d'identification automatique radiofréquences se compose d'un lecteur qui transmet un signal selon une fréquence déterminée vers une ou plusieurs étiquettes radio situées dans son champ de lecture. Celles-ci transmettent en retour un signal. La figure suivante (**Figure II.1**) montre le fonctionnement général d'un système RFID.

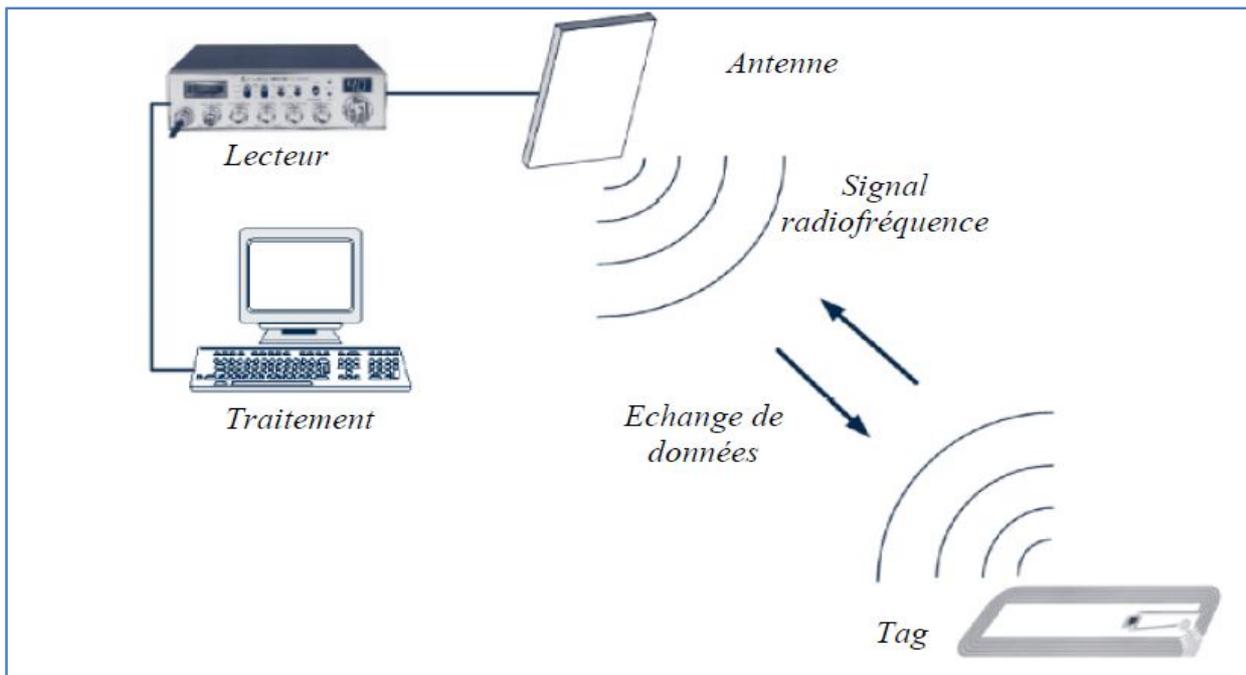


Figure II.1: Fonctionnement d'un système RFID. [10]

II.4 Les composants d'un système RFID :

Un système RFID est constitué essentiellement de trois éléments (**Figure II.2**) : les transpondeurs ou les étiquettes (**Tags**) qui stockent un identifiant (**ID**), les lecteurs ou les interrogateurs qui récupèrent les identifiants des étiquettes, et le Middleware qui interprète les informations fourni par les lecteurs pour des applications spécifiques [11].

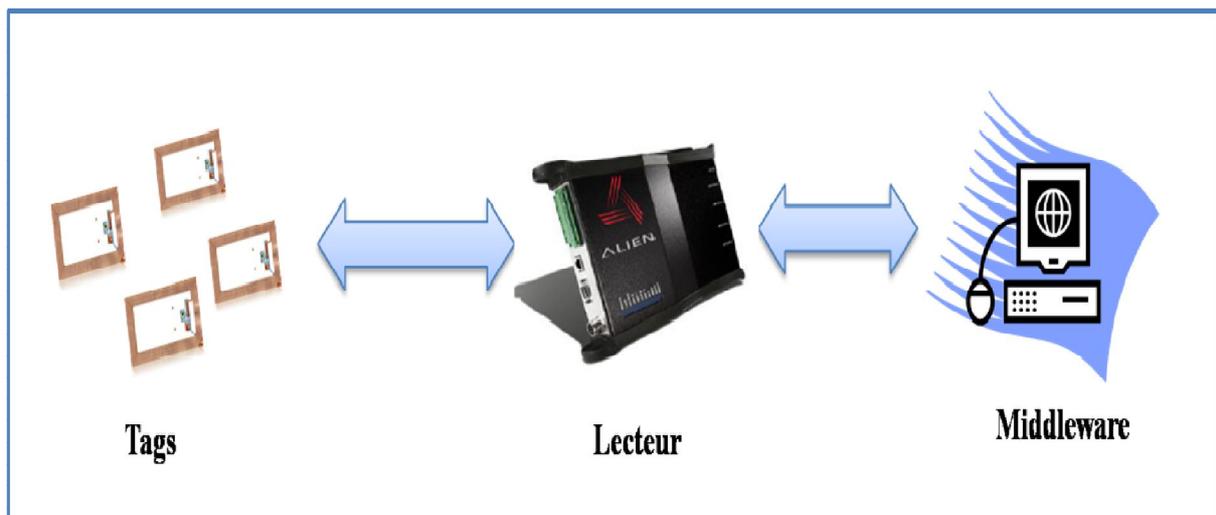


Figure II.2: Les composants d'un système RFID [11].

II.4.1 Les lecteurs RFID (Interrogateurs) :

Ce sont des dispositifs actifs, émetteurs des signaux RF (radiofréquences) qui vont activer les tags qui passent devant eux en leur fournissant l'énergie dont ils ont besoin pour fonctionner.

Outre de l'énergie pour l'étiquette, le lecteur envoie des commandes particulières auxquelles répond le tag. L'une des réponses les plus simples possibles est le renvoi de l'identifiant du tag. [12]

Les interrogateurs se présentent généralement sous la forme d'un boîtier électronique que l'on relie à l'antenne externe, mais aussi au système informatique par liaison filaire ou radio.

Les lecteurs peuvent être fixes ou mobiles, et peuvent prendre plusieurs formes, par exemple s'intégrer dans le cadre d'une porte, pour une application de contrôle d'accès.



Figure II.3: Des lecteurs RFID.

II.4.1.1 Les composants d'un lecteur RFID:

Le lecteur RFID est l'interface entre une application hôte et les tags RFID. Son rôle est de réaliser la gestion de la communication avec les tags RFID et de transmettre leurs données à l'application hôte [10]. Un lecteur est composé de différentes unités illustrées sur la (Figure II.4).

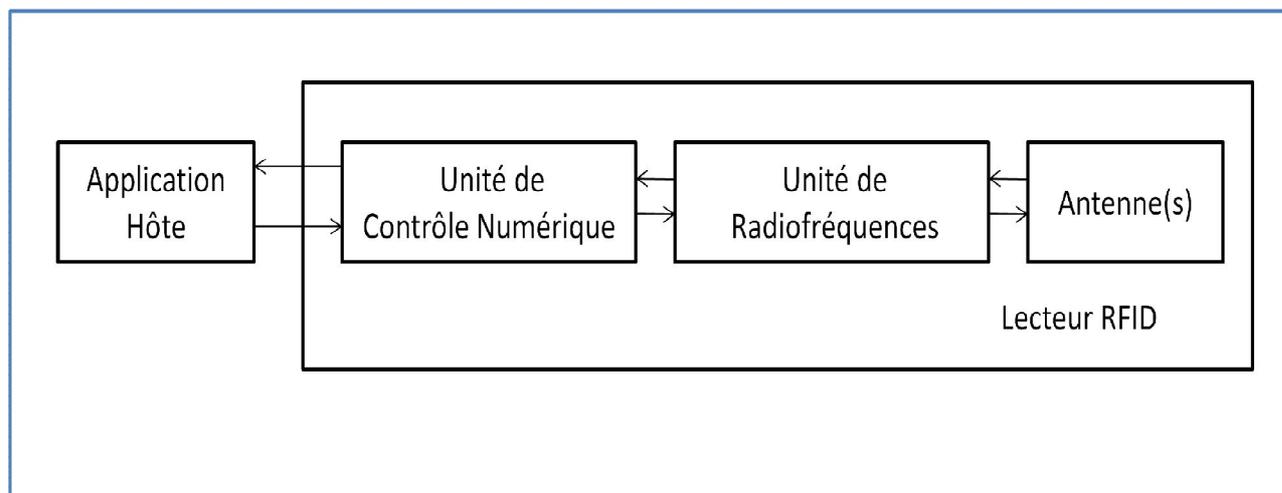


Figure II.4: Les composants d'un lecteur RFID. [10]

1. **Unité de Contrôle Numérique** : Composée d'un microprocesseur, d'une mémoire et d'une interface de commandes pour communiquer avec l'application hôte. C'est donc cette unité de contrôle qui génère et met en forme le signal numérique contenant l'information à transmettre au tag et traite en retour, la réponse de celui-ci. Ansaï, en plus de l'implémentation de protocole de communication, l'unité numérique est en charge du codage et du décodage des signaux et éventuellement du cryptage et du décryptage des données ou toute autre fonction imposée par l'application hôte [13].
2. **Unité de Radiofréquences** : Constituée d'un émetteur et d'un récepteur des signaux radiofréquences. Cette partie est en charge de la génération d'une porteuse radiofréquence, de sa modulation avec un signal numérique généré par l'unité de contrôle numérique et de la démodulation de la réponse des tags [10].
3. **Antenne(s)** : Permettant de transmettre et de recevoir les données, et de propager l'énergie radiofréquence télé-alimentant les tags.

II.4.1.2 Gammes de fréquences :

Actuellement quatre gammes de fréquences radio sont principalement utilisées Dans un lecteur RFID [11]. Leur différence tient surtout dans la distance et la vitesse de lecture.

- **La basse fréquence (BF)**: de 125 kHz permet une lecture seule de 20 à 50 cm.
- **La haute fréquence(HF)** : de 13,56 MHz associe soit : une grande vitesse à une petite distance, soit une vitesse moyenne à une distance jusqu'à 70cm.
- **L'ultra haute fréquence (UHF)** : de 800 à 950 MHz avec des lectures de 2 à 5 mètres.
- **L'hyper fréquence(SHF)** : de 2,4 à 5,8 GHz pour des grandes distances.

II.4.2 Les tags RFID :

Ce sont des dispositifs récepteurs, que l'on place sur les éléments à identifier (objet, animal...). Ils sont munis d'une puce contenant les informations et d'une antenne pour permettre les échanges d'informations avec le lecteur. [12]



Figure II.5: Des tags RFID.

II.4.2.1 Les composants d'un tag RFID :

L'étiquette RFID se compose d'une puce électronique et d'une antenne, qui sont montées sur un support plastique ou papier (Figure II.7).

1. La puce électronique : Possède trois composants (Figure II.6).

1. **Une Interface Radiofréquence** : Pour réceptionner les signaux RF et échanger des données avec l'interrogateur [14].
2. **Un Séquenceur Numérique** : Pour analyser les instructions reçus de l'interrogateur, codé/décoder les informations, et répondre en envoyant ces données à l'interface radiofréquence [13].
3. **Une Mémoire non volatile** : Pour le stockage de l'identifiant unique. La mémoire non volatile a parfois une capacité supérieure de la taille du l'identifiant. Cela permet de stocker des informations complémentaires sur l'objet au cours de son cycle de vie, cette partie de la mémoire est en lecture et écriture [14].

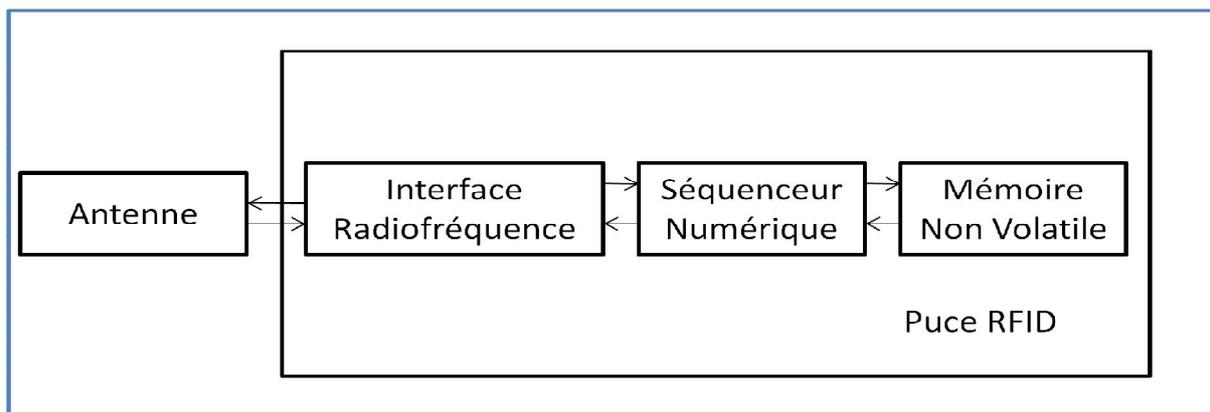


Figure II.6: Les composants d'une puce RFID. [10]

2. **L'antenne :** La fonction fondamentale d'une antenne est d'absorber les ondes RF et de renvoyer le signal. L'antenne alimente la puce en puisant l'énergie du champ électromagnétique.
3. **Le support :** En général, la puce est connectée à l'antenne extérieure à l'aide de deux plots. Le tout est souvent monté sur un premier support en général souple appelé inlay. Selon l'utilisation, cet inlay est ensuite intégré dans un support plastique ou sur un support papier [14].

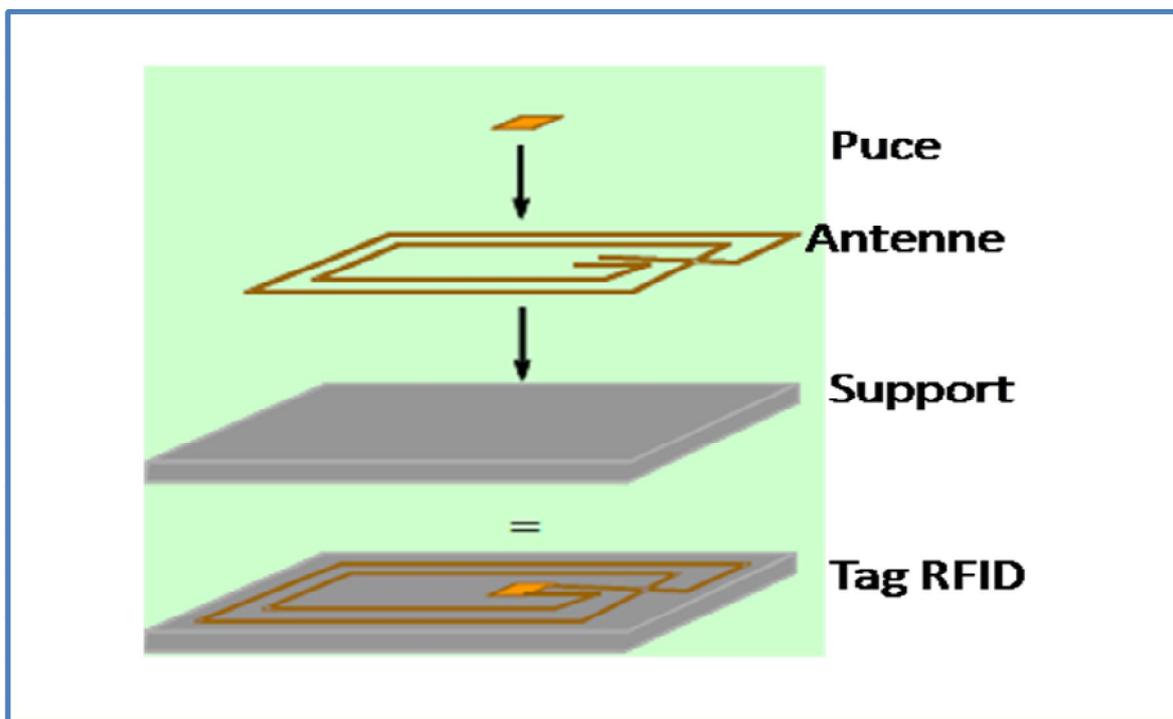


Figure II.7: Eléments constitutifs du tag RFID.

II.4.2.2 Types des tags RFID :

Selon la nature de la source d'alimentation énergétique il existe trois principaux types d'étiquettes RFID [15]:

a) Les étiquettes passives :

Ce sont les tags les plus courants. Ils ne possèdent aucune source d'alimentation interne et doivent donc être alimentés par une source extérieure. C'est donc le lecteur qui alimente les tags présents dans son champ par un procédé que l'on nomme "**Télé-alimentation**". Les tags passifs sont utilisés lorsqu'il n'est pas nécessaire qu'ils soient alimentés en permanence.

Télé-alimentation : Le principe de télé-alimentation repose sur le phénomène d'induction électromagnétique. Voici donc un petit rappel : Il a été découvert que lorsqu'on appliquait un courant électrique à une bobine de cuivre, alors un champ magnétique était induit. Par réciproque, lorsqu'une bobine de cuivre est parcourue par un champ magnétique, alors un courant électrique est induit à ses bornes. C'est ce que l'on appelle l'induction électromagnétique. Tout le mécanisme de communication en RFID repose sur ce principe. Les antennes utilisées pour les tags et les lecteurs de base sont des bobines de cuivre. Les circuits de lecteur, dont l'antenne, sont alimentés par son environnement (hôte). De ce fait, un champ magnétique est induit et propagé par l'antenne du lecteur. Lorsqu'un tag entre dans le champ magnétique d'un lecteur, son antenne de cuivre est parcourue par ce champ. Ses circuits sont alors alimentés par le courant électrique induit. Les deux appareils sont donc en mesure de communiquer.

b) Les étiquettes actives :

A l'inverse de celle passive, l'étiquette active possède sa propre source d'énergie. Cette source d'énergie peut être solaire ou sous forme de batterie. L'avantage de ces étiquettes est qu'elles transmettent leur information en continu et elles ne nécessitent pas la présence d'un lecteur pour fonctionner. Un mode « veille » existe sur ce type d'étiquette afin d'économiser la source d'énergie. Elles ont de plus comme avantage de pouvoir être lues à une distance plus grande. Désavantage de la technologie, la multiplication des ondes dans un environnement peut créer des interférences. En effet, si la puce communique en continu, le nombre de message dans un entrepôt peut être trop important.

c) Les étiquettes semi-actives :

Ces étiquettes sont bien sûr à mi-chemin entre les deux technologies vues ci dessus. L'étiquette possède une source d'énergie propre mais nécessite un lecteur pour émettre les données qu'elle contient. L'avantage de cette technologie par rapport à une

étiquette passive est qu'elle sera lue beaucoup plus vite et surtout à une distance plus longue.

II.4.2.3 Les différentes utilisations des tags RFID :

Il existe 3 types d'utilisation du tag RFID [9]:

a) **Les étiquettes à lecture seule :**

La lecture seule est la technologie la plus simple. Elle consiste en une étiquette contenant des données qui peuvent être lues par un lecteur mais sans possibilité de les modifier. L'étiquette est bloquée dès sa fabrication. Ce sera donc l'entreprise qui la fabrique qui entrera les données dans la puce et qui bloquera cette dernière. Ce type d'étiquette s'adresse donc plutôt à de grands volumes et à une utilisation de masse comme dans la grande distribution.

b) **Les étiquettes à écriture unique et lecture multiple:**

Les étiquettes WORM (Write Once, Read Many) sont des étiquettes fournies vides sans aucune information. Les utilisateurs ont la possibilité de graver les informations sur l'étiquette qu'une seule fois, mais ils peuvent avoir recours à ces données ou informations pour lecture à plusieurs reprises.

c) **Les étiquettes à lecture et écriture multiple:**

Les utilisateurs reçoivent des étiquettes vierges. Ils ont la latitude de graver et lire les informations à plusieurs reprises. Par conséquent ces étiquettes peuvent être lues, effacées, modifiées plusieurs fois. Ceci est un avantage dans plusieurs applications où le code d'identification nécessite de changements et où les données variables sont plus importantes qu'une identité unique.

On peut donc voir que la technologie des puces d'identification par radiofréquence est étendue et permet de s'adapter à des situations différentes. De plus, la recherche dans le domaine de la RFID est en constante évolution et apportera donc dans les années à venir des améliorations continues notamment dans le domaine de la fiabilité, de la miniaturisation et de l'autonomie.

II.4.3 Le middleware: [16]

Le middleware RFID est un intermédiaire entre la technologie RFID et le système d'informations d'entreprise. Les technologies RFID peuvent fournir une quantité massive d'informations. Cependant, les systèmes d'informations d'entreprises veulent obtenir seulement les données nécessaires et précises. Donc le middleware filtre les données excessives et inutiles, assurant l'exactitude de ces données, les transforme à une information utile pour améliorer leur gestion. Les fonctions du middleware sont présentées sur la (Figure II.7).

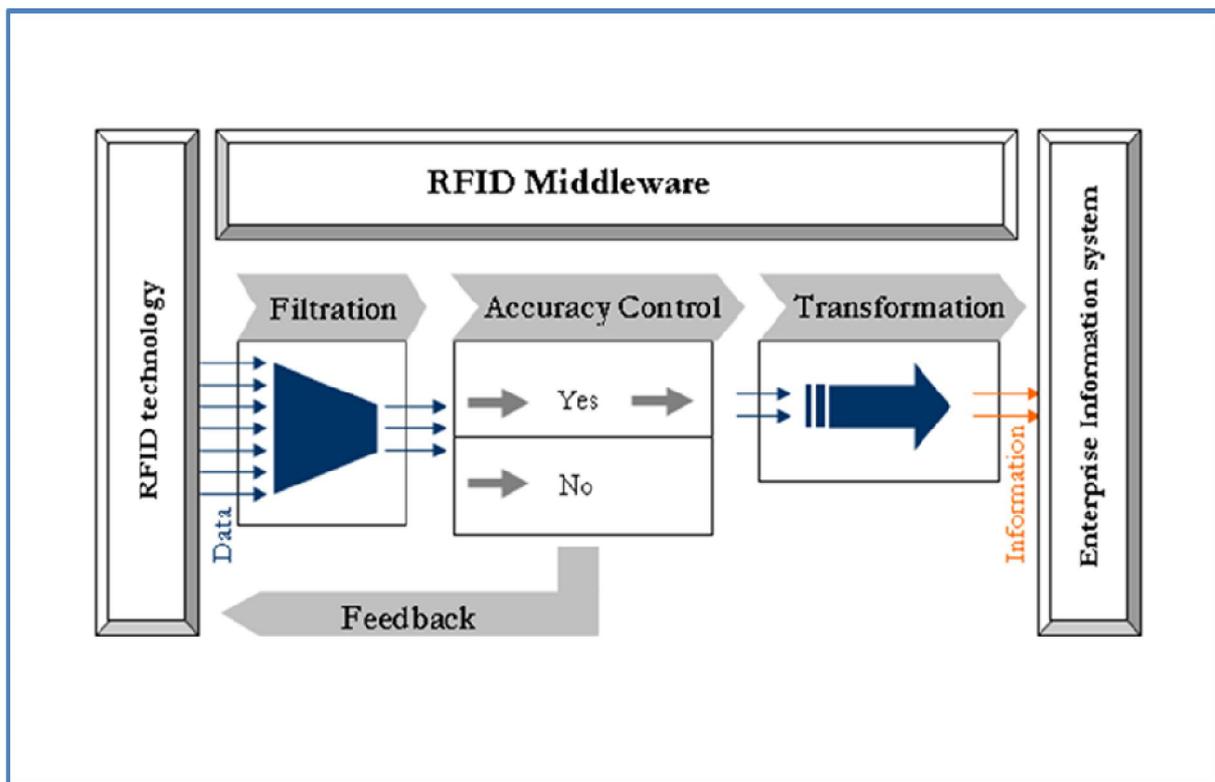


Figure II.8: Les fonctions du middleware RFID. [16]

II.5 Communication Tag passif ↔ Lecteur :

Nous avons vu que les tags passifs sont télé-alimentés. La communication consiste donc en un transfert de données associé à un transfert d'énergie. La communication des données est bidirectionnelle : la communication du lecteur vers le tag est appelée liaison montante (uplink) et la réponse du tag vers le lecteur est appelée descendante (downlink). La figure suivante illustre le principe d'une communication RFID. (Tags passifs) (Figure II.9).

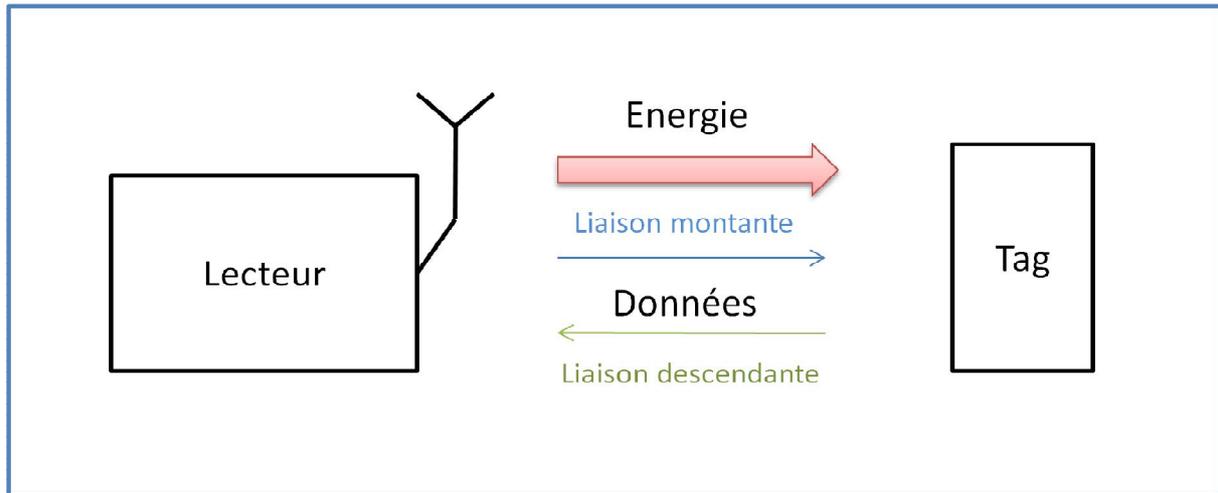


Figure II.9: Représentation schématique d'une communication RFID.

II.5.1 Nature de la communication: [15]

Les communications RFID, comme dans la plupart des communications sans fil, sont half-duplex. Cela signifie que chacun des interlocuteurs (ici le lecteur et le ou les transpondeurs) communiquent à tour de rôle. Il existe cependant deux modes de communication liés à la présence des deux types d'information transmis (données et énergie) :

- **Simultané** : Les données et l'énergie sont transmises simultanément au transpondeur.
- **Non simultané** : Les données et l'énergie sont fournies alternativement au transpondeur.

Comme dans toute conversation, l'un des deux interlocuteurs doit nécessairement initialiser la communication. Pour cela, il existe deux modes :

- **TTF (Tag Talks First)** : Dans ce mode, le tag annonce sa présence à son arrivée dans le champ d'un lecteur. Ce mode peut poser des conflits lorsque plusieurs tags annoncent leur présence simultanément.
- **RTF (Reader Talks First)** : Dans ce mode, le lecteur interroge constamment son environnement afin de détecter la présence de nouveaux arrivants. Une requête est propagée régulièrement et, lorsqu'un transpondeur entre dans le champ et est capable de répondre, il renvoie une réponse annonçant sa présence.

II.5.2 Codage des signaux : [8]

Le codage des signaux binaires générés par la partie numérique du tag ou du lecteur est la première étape dans la préparation à la communication en RFID. Par symétrie, la phase de décodage des signaux est la dernière étape. Les algorithmes de codage et décodage des signaux sont, bien évidemment, symétriques. Plusieurs types de codages sont utilisés : RZ (Retour à Zéro), NRZ (Non RZ), Miller ou Manchester, etc.

Pour obtenir les performances souhaitées le choix du codage doit permettre :

- **Pour la liaison montante :**

- D'avoir une bonne efficacité de transfert d'énergie.
- De laisser du temps de travail et calcul au tag pendant la transmission.
- D'assurer un bon rapport signal sur bruit.

- **Pour la liaison descendante :**

- De minimiser la consommation énergétique pendant son fonctionnement.
- De faciliter la détection par le lecteur, même en présence de bruit et en présence d'autres tags.

II.5.3 Modulation :

La modulation des signaux est la seconde étape dans la préparation à la communication en RFID. Par symétrie, la phase de modulation implique une phase de démodulation des signaux à la réception de signaux RF de réponse. Les méthodes de modulation et démodulation des signaux sont, bien évidemment, symétriques.

L'intérêt de la modulation est de pouvoir transmettre le signal fréquentiel de données. Pour que le lecteur puisse transmettre un message (signal binaire généré par l'unité numérique) à un transpondeur, il doit d'abord le moduler avec une porteuse. La porteuse est un signal de haute fréquence que nous allons moduler selon des techniques différentes, mais qui conduiront au final à la transmission du message [15].

En RFID, les dispositifs qui communiquent ne sont pas technologiquement conçus de la même façon. Pour cette raison, des types de modulation différents sont utilisés selon le sens de la communication :

- **Pour la liaison montante :**

Les deux techniques les plus largement utilisées sont la modulation d'amplitude (ASK=Amplitude Shift Keying) et la modulation de fréquence (FSK =Fréquence Shift Keying). Ces deux techniques présentent l'avantage de permettre une démodulation facile à réaliser, ce qui permet d'alléger l'architecture du tag [13].

- **Pour la liaison descendante :**

Le transpondeur ne peut pas se comporter comme un émetteur de signaux RF. En effet, il ne dispose pas, dans son interface RF, des mécanismes permettant d'émettre un signal radiofréquence vers le lecteur. Les transpondeurs utilisent ce qu'on appelle la réflexion d'ondes de la porteuse provenant de lecteur. Pour cela, les tags utilisent une modulation différente que l'on appelle OOK (On Off Keying) correspond à une modulation d'amplitude "tout ou rien". La modulation OOK agit un peu comme un interrupteur : la valeur binaire '0' correspond à 0V (on ne laisse passer aucun signal), la valeur binaire '1' permet de laisser passer le signal tel quel [15].

II.5.4 Protocole de communication: [10] [13]

Le protocole de communication entre le lecteur et le tag est généralement composé de trois phases temporelles :

- **Phase de réveil du tag :** Le lecteur envoie une onde électromagnétique vers le tag pour lui permettre de récupérer l'énergie nécessaire à son fonctionnement et de se mettre dans un état d'attente des instructions à venir de lecteur.
- **Phase d'instruction :** Le lecteur envoie un message au tag. C'est une transmission numérique sur porteuse qui assure cette communication. Cette communication se traduit en réalité par une variation de l'onde électromagnétique envoyée. Or, parallèlement à l'envoi des instructions, le lecteur doit continuer à assurer l'alimentation du tag. Ainsi un compromis doit être trouvé lors de la mise en forme du signal envoyé par le lecteur afin d'assurer ces deux fonctions. Cette mise en forme nécessite donc un choix judicieux du codage de l'information et de la technique de modulation.
- **Phase de lecture :** Le tag envoie sa réponse au lecteur. Là encore, un compromis doit être trouvé entre ce transfert de données descendant et la puissance que le tag doit continuer à absorber afin d'assurer son alimentation.

Le protocole de communication est contrôlé au niveau du lecteur et du tag par la partie numérique qui reçoit et transmet les données par une interface radiofréquence.

II.6 Les fréquences d'un système RFID: [16]

Le dialogue entre le tag et le lecteur est régi par un protocole de communication dont la principale caractéristique est la fréquence radio d'échange. Plusieurs fréquences de communication cohabitent au sein de la technologie RFID, les principales sont présentées dans le tableau suivant :

Fréquences	Type de tag	Caractéristique	Applications
Basse fréquence(BF) 125Khz	Passif	<ul style="list-style-type: none"> • Distance de lecture moyenne (10 à 150 cm). • Rapidité de lecture moyenne. 	<ul style="list-style-type: none"> • Contrôle d'accès • Antivol, anti-démarrage
Hautes Fréquences(HF) 13,56 Mhz	Passif	<ul style="list-style-type: none"> • Distance de lecture faible Quelques centimètres. • Rapidité de lecture faible. 	<ul style="list-style-type: none"> • Contrôle d'accès • Paiement • Bibliothèque
Ultra Hautes Fréquences(UHF) 900 Mhz	Passif Actif	<ul style="list-style-type: none"> • Grande distance de lecture jusqu'à 5 mètres. • Vitesse de lecture importante. 	<ul style="list-style-type: none"> • Logistique, gestion de stocks multiples.
Hyper Fréquences(SHF) 2,4 Ghz	Actif Semi-actif	<ul style="list-style-type: none"> • Très grande distance de détection (>10 mètres). • Très grande vitesse de lecture. 	<ul style="list-style-type: none"> • Logistique militaire. • Géo-localisation.

Tableau II.1: Plage des fréquences d'un Système RFID.

II.7 La standardisation RFID : [16]

Afin d'obtenir des performances maximales de la technologie RFID ; les étiquettes, lecteurs et fréquences doivent fonctionner en harmonie totale même en cas de fabricants différents, à cette effet plusieurs organisations ont établies des standards internationales

permettant le développement considérable d'applications basés sur cette technologie. On peut distinguer deux types de standards:

III.7.1 Les standards de données :

Ils définissent le contenu des mémoires des étiquettes RFID, ainsi que le format de stockage. EPC (Code Produit Electronique) est l'un des standards de données. Il a été développé par EPCGlobal afin d'assurer une identification des objets d'une manière unique dans le monde. Un EPC est un code unique pour chaque produit, il peut contenir plusieurs informations telles que le fabricant et le type du produit...etc. Actuellement, il existe trois formats d'EPC : 64 bits, 96 bits ou 128 bits. La figure suivante présente un EPC-96 bits.

Norme de codage	Entreprise	Type de produit	Numéro de série du produit
01	0000A89	00016F	000169DC0

Figure II.10: Exemple d'un EPC-96 bits.

II.7.2 Les standards de technologie :

Ils se focalisent sur les protocoles de communication entre le tag et le lecteur. L'une des organisations qui travaille sur les standards de technologie est l'ISO (International Organization for Standardization). Grace à son partenariat avec des organismes internationaux, des gouvernements, des industries et des représentants des consommateurs, elle a fourni des standards internationaux dans environ 148 pays. Plusieurs normes ont été développées par l'ISO, le tableau suivant présente quelques normes de l'ISO.

Les standards ISO	Caractéristiques
ISO 11785	Pour des applications BF
ISO 15693	Pour des applications HF
ISO 18000-6	Pour des applications UHF
ISO 15693	Pour des tags Passifs, Lecture/Ecriture et qui travaillent avec HF

Tableau II.2: Quelques standards ISO.

II.8 Applications RFID :

Les applications RFID s'appuient sur différents standards dépendant des fonctionnalités exigées par les processus métier et par certaines contraintes locales.

La technologie RFID permet de répondre à un grand nombre de besoins. Elle est utilisée dans divers domaines comme le montre le tableau suivant :

Domaine d'application	L'utilisation
Sécurité	<ul style="list-style-type: none"> - Gestion du personnel. - Contrôle d'accès aux zones réservées. - Authentification d'objet.
Industrie	<ul style="list-style-type: none"> - Identification et suivi de vêtements. - Blanchisserie industrielle. - Identification et suivi des bouteilles de gaz.
Agroalimentaire	<ul style="list-style-type: none"> - Suivi de la chaîne du froid des produits alimentaires. - Suivi de la chaîne de fabrication des produits frais. - Suivi du bétail.
Logistique	<ul style="list-style-type: none"> - Suivi de bagages dans le transport aérien. - Suivi de sacs postaux. - Suivi et pistage de containers. - Identification de produits palettisés. - Contrôle d'accès (parking ...).
Médical	<ul style="list-style-type: none"> - Gestion de collectes des déchets médicaux jusqu'à l'incinération. - Tatouages électroniques pour animaux.
Véhicule	<ul style="list-style-type: none"> - Gestion de flotte de véhicules. - Authentification de véhicule. - Paiement des carburants. - Antivol, anti-démarrage. - Contrôle des pneumatiques.
Loisirs	<ul style="list-style-type: none"> - Location de K7 vidéo et DVD. - Bibliothèque (gestion rapide des entrées/sorties). - Ticketing (remontées mécaniques dans les stations). - Gestion des temps des coureurs.

Tableau II.3: Exemples d'applications RFID.

II.9 Avantages et inconvénients de la RFID : [9]

II.9.1 Avantage :

➤ **Possibilité de modification de données**

Pour les étiquettes à lectures et écritures multiples, les données gravées peuvent subir des à tout moment par les personnes autorisées. Ce qui peut permettre la mise à jour de données dans une étiquette favorisant ainsi la traçabilité, le suivi en temps réel.

➤ **Grand volume de données**

Les étiquettes RFID peuvent contenir de données dont les caractères peuvent aller jusqu'à plus de 15000 caractères.

➤ **La vitesse de marquage**

Les étiquettes RFID peuvent être incorporées dans le support de manutention ou les emballages, et les données concernant les produits sont introduites ou modifiées en une fraction de seconde au moment de la constitution de groupage logistique par les serveurs (ordinateurs), ce qui permet aussi une grande vitesse de lecture.

➤ **Protection des contenus**

Les contenus des étiquettes RFID peuvent être protégés par un mot de passe en lecture ou écriture. Avec cette protection contre l'accès des informations imprimées sur l'étiquette, la contrefaçon et le vol s'avèrent difficiles.

➤ **Durée de vie**

Les étiquettes RFID peuvent avoir une durée de vie de dizaines d'années. Les données au cours de ces années peuvent subir de modifications plus d'un million de fois selon le type de l'étiquette avec un maximum de fiabilité.

➤ **Meilleure accessibilité et résistante aux effets extérieurs**

Les étiquettes de la technologie RFID fonctionnant avec les ondes électromagnétiques n'ont pas besoin de contact ou de visée optique. Leur liaison avec le système est établie dès qu'elles entrent dans les champs électromagnétiques.

Ainsi, les personnes, les produits et équipements pourvus des étiquettes RFID se trouvant dans le rayon couvert par le champ électromagnétique du lecteur peuvent être sujets à une traçabilité en temps réel et de manière permanente. Les étiquettes RFID sont insensibles à la poussière, aux taches, aux frottements, à l'humidité. En plus la lecture peut être effectuée en vrac, permettant la lecture simultanée de plusieurs étiquettes.

II.9.2 Inconvénients :**➤ Prix**

Le coût d'implantation d'un système RFID est relativement élevé, ce qui suscite des préoccupations concernant le retour sur investissement. Les prix des étiquettes RFID varient entre 25 cents et 250\$ et ceux de lecteurs coûtent de centaines de dollars, ce qui est largement supérieur au seuil de rentabilité pour les fabricants de produits à faibles coûts. Il faut aussi prendre en considération le coût élevé du logiciel qui devra régir l'application.

➤ Interférence des ondes

Les informations et les données gravées sur une étiquette peuvent être sujettes à des interférences des ondes entre elles. Et dès que plusieurs étiquettes se retrouvent dans le champ de lecture, les données sont saisies par le lecteur en même temps. Les études sont en cours de finalisation pour la parcellisation et le groupage des fréquences en fonction des applications, et pour assurer l'unicité de captage des informations uniquement par le lecteur autorisé.

➤ Perturbations métalliques

La lecture des étiquettes RFID peut aussi être perturbée par la proximité dans le champ électromagnétique des éléments métalliques ce qui affecterait fortement la réussite de la technologie dans le domaine de production métallique.

➤ Les préoccupations sur la vie privée

Les systèmes RFID, s'ils peuvent simplifier la vie, sont autant de menaces pour le respect de la vie privée et des libertés individuelles. Par exemple, les cartes de transports publics, laissent des traces du passage de l'individu dans le système informatique. Ainsi, l'administrateur du système et d'autres personnes encore, autorisées ou non, peuvent potentiellement avoir accès à l'historique des déplacements d'un individu : où était-il ? À quelle heure ? Où allait-il ?

II.10 La technologie NFC

Identifier c'est bien mais communiquer c'est mieux, et c'est dans cette optique qu'est née le NFC sous l'impulsion du NFC Forum en 2004, principalement conduit par Nokia. Il est basé sur le même principe de fonctionnement que le RFID.

La technologie NFC décrit un moyen de communication sans contact pour les appareils mobiles tels que les Smartphones. La connexion radio se fait en rapprochant deux périphériques l'un de l'autre (quelques centimètres) sans paramétrage préalable.

II.10.1 Définition :

La communication en champ proche (en anglais *Near Field Communication*, **NFC**) est une technologie de communication sans-fil à courte portée et haute fréquence, permettant l'échange d'informations entre des périphériques jusqu'à une distance d'environ 10 cm. Cette technologie est une extension de la technologie RFID. L'objectif initial était de combiner la fonction lecteur et la fonction transpondeur dans un même dispositif [8].

La plupart des derniers Smartphones Android, Windows Phone ou encore BlackBerry sont compatibles et certifiés NFC.

II.10.2 Modes de fonctionnement du NFC: [8]

La NFC a trois modes de fonctionnement différents :

II.10.2.1 Mode émulation de carte :

La NFC permet de faire fonctionner un terminal mobile (ex : Smartphone) en mode émulation de carte, c'est à dire que le terminal mobile émule le fonctionnement d'une carte sans-contact (tag) devant un lecteur RFID. Dans le cas où le terminal mobile est un téléphone mobile compatible GSM ou UMTS, la carte SIM est utilisée comme élément de sécurité.

Exemples d'utilisations :

- Paiement en utilisant un appareil mobile (téléphone portable, Smartphone, ordinateur portable, tablette numérique...) sur un terminal de paiement sans contact .
- Accès et démarrage d'un véhicule à l'aide de son téléphone mobile.

II.10.2.2 Mode lecteur :

Le terminal mobile devient un lecteur de cartes sans-contact ou de tags passifs. Ce mode permet à un usager de lire ou écrire des informations en approchant son mobile devant ces étiquettes.

Exemple d'utilisation :

- Lecture d'une carte de visite électronique avec un Smartphone.

II.10.2.3 Mode pair-à-pair :

Ce mode permet à deux terminaux mobiles d'échanger de l'information, par exemple des photos, des vidéos, etc. Un appareil doté de la technologie NFC est capable d'échanger des informations avec des tags mais également avec d'autres appareils dotés de cette technologie.

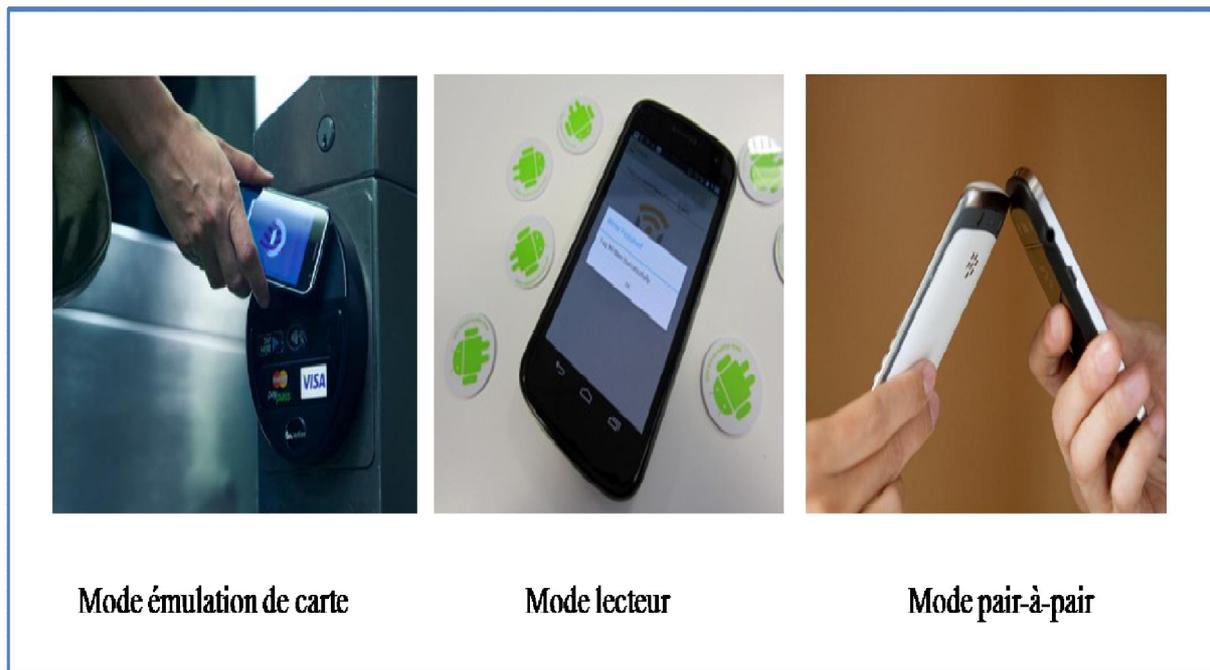


Figure II.11: Les modes de fonctionnement du NFC.

II.10.3 NFC et Android: [17]

La plupart des derniers Smartphones Android embarquent un capteur NFC. Le SDK (Software Development Kit) doté de deux nouvelles mises à jour (API 9 et 10 pour Android 2.3 Gingerbread) permette de manipuler ce capteur. Bien que le capteur permette (au niveau physique) les trois modes du NFC (à savoir la lecture, pair-pair et l'émulation), actuellement seuls les deux premiers sont supportés par le SDK.

Conclusion :

Aujourd'hui, la technologie RFID fait une partie du quotidien d'un nombre croissant d'entreprises et de filières (élevage, antivol textile, transport public, construction automobile, etc.). Dans ce chapitre nous avons présenté d'une manière générale cette technologie, son principe de fonctionnement, les plages de fréquence utilisées, ses équipements, ses applications, ses avantages et inconvénients, et enfin la technologie NFC que nous avons utilisé pour la réalisation de notre application. Le chapitre suivant sera consacré pour détailler la phase conception de notre application.

Introduction

Après avoir pris connaissance dans les chapitres précédents du principe de la technologie RFID ainsi que la technologie NFC, nous présentons dans ce chapitre le processus de développement de notre application. Nous allons tout d'abord présenter les objectifs de notre projet puis l'architecture de notre système. Ensuite, nous allons expliquer chacun de ses composants. Enfin, nous décrirons d'une manière détaillée le fonctionnement de notre application.

III.1 Présentation et objectifs du projet

III.1.1 Présentation

Afin de rendre la vie des campus encore plus facile, de nombreuses universités dans le monde ont choisi la carte multiservices pour faciliter l'accès des étudiants aux nombreux services universitaires. Elle est à la fois une carte d'étudiant, une carte de bibliothèque et une carte de cité universitaire... etc.

La carte multiservices est munie d'une étiquette RFID permettant une identification de la carte au moyen de lecteurs de cartes à puce sans contacts. Cette caractéristique permet au détenteur de s'identifier électroniquement dans le cadre de services.

La carte multiservices permet :

- ✓ De contrôler l'accès aux locaux de l'université (salle machine, laboratoire, cité universitaire...)
- ✓ D'emprunter des ouvrages de la bibliothèque.
- ✓ De payer les repas au restaurant universitaire.
- ✓ De pratiquer les activités sportives.

Dans notre projet nous avons essayé d'implémenter deux services pouvant être fournis par RFID en utilisant un Smartphone compatible avec la technologie NFC qui joue le rôle de lecteur de cartes, et un ensemble des tags RFID simulant les cartes multiservices.

III.1.2 Objectifs :

En se basant sur la technologie NFC (mode lecteur), notre projet consiste à concevoir et réaliser une application Android permettant :

- D'identifier un étudiant grâce à l'étiquette RFID collée sur sa carte multiservices et d'afficher ses informations sur un Smartphone NFC, cette identification permet par exemple à un agent équipé de ce Smartphone de contrôler l'accès à des locaux en accès limité (salle machines, laboratoire... etc.).
- Une gestion intelligente de la bibliothèque afin de faciliter l'emprunt et le retour des livres. Dans ce cas tous les livres ont une étiquette RFID pour une identification rapide. Equipé d'un Smartphone NFC, le bibliothécaire, près de l'entrée, gère l'accès et enregistre les emprunts et les retours des livres.

III.2 Description du système

Le but de notre application étant d'identifier des étudiants par un Smartphone NFC grâce à leurs tags, il était nécessaire d'avoir recours à un système de base de données. En effet, seul un ID peut être enregistré sur le tag. Il faut donc un système permettant de stocker et de restituer les informations de l'étudiant correspondant à cet ID. Et puisque le Smartphone n'a pas d'espace mémoire suffisant pour héberger une base de données contenant toutes les informations des étudiants, notre choix s'est porté sur une architecture client/serveur dans laquelle la base de données sera située sur un serveur de base de donnée.

Notre système RFID repose sur une architecture client-serveur à trois niveaux (**Figure III.1**) :

- **Le client mobile (niveau 1) :** Représente une application Android qui s'exécute sur un Smartphone compatible avec la technologie NFC, elle est responsable de la manipulation des tags RFID (Lecture/Ecriture) et de la transmission et la réception des données vers/depuis le serveur.
- **Le serveur d'application (niveau 2) :** Il est chargé de répondre aux requêtes de client mobile afin d'offrir les services demandés, mais en faisant appel à un autre serveur.
- **Le serveur de base de données (niveau 3) :** Il stocke et gère les données des utilisateurs (étudiants, administrateurs), et il les fournit au serveur d'application.

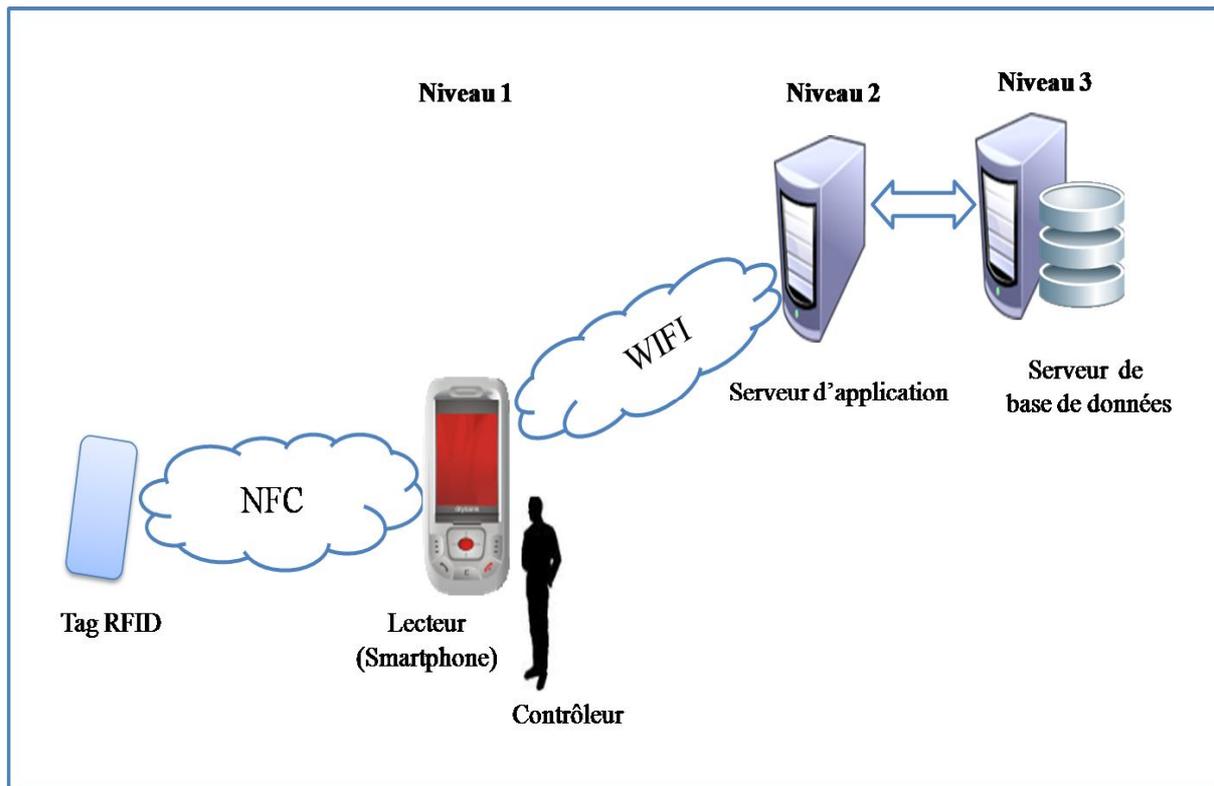


Figure III.1: Architecture générale de notre système.

III.3 Description des composants du système

Dans cette partie, nous décrivons d'une manière détaillée chacun des composants du système tout en montrant les liens existant entre eux.

III.3.1 Les Tags RFID

L'étiquette RFID est le cœur de notre système, elle est utilisée pour identifier au niveau du lecteur (Smartphone) l'objet sur laquelle elle est collée. Dans notre système on la trouve sur :

- **Les cartes multiservices:** Pour identifier l'étudiant porteur de cette carte d'une manière unique, l'étiquette RFID va contenir le matricule d'inscription de l'étudiant.
Exemple : 14D030772013.
- **Les livres :** Dans ce cas le tag contient la cote de livre.
Exemple : IF120.

Ces informations (matricule, cote) sont introduites dans les tags par une application Android implémenté au niveau du lecteur (Smartphone), en utilisant la technologie NFC pour les transmettent.

III.3.2 Le Lecteur

Le lecteur assure la communication et l'affichage d'information dans notre système, il représente un Smartphone compatible avec NFC et doté d'une liaison WIFI. Selon le type de la technologie de communication utilisée (NFC ou WIFI) on peut distinguer deux fonctions principales effectuées par notre lecteur :

1. Manipulation des tags :

En utilisant la technologie NFC, le lecteur interroge les tags afin de récupérer l'identifiant qu'ils contiennent, mais aussi, il permet à l'administrateur d'écrire des données (matricule d'inscription, cote de livre) dans la mémoire des tags.

2. Echange des données avec le serveur d'application :

Lorsque le lecteur reçoit l'identifiant du tag qui se trouve dans son champ de lecture NFC, il l'envoie dans une requête au serveur d'application à travers une connexion sans fil WIFI. Le serveur d'application traitera à son tour la requête correspondante en faisant appel au serveur de base de données et renverra la réponse au lecteur.

Les requêtes envoyées par le client mobile vers le serveur d'application ont la forme : **[Identifiant_Tag, Code_Service]**, elles permettent de solliciter des services au niveau du serveur, on trouve :

- [Matricule_Etudiant, "Contrôle_Accès "] : Pour récupérer les informations de l'étudiant correspondant à cet matricule afin de contrôler l'accès de cet étudiant au campus.
- [Matricule_Etudiant, "Accès_Bibliothèque "] : Pour récupérer les informations de l'étudiant correspondant à cet matricule afin de contrôler l'accès de cet étudiant aux services de la bibliothèque.
- [Matricule_Etudiant, "Livre "] : Pour récupérer la liste de tous les livres empruntés et resitués par l'étudiant qui porte cette carte.
- [Cote_Livre, "Emprunter "] : Pour enregistrer l'emprunt de ce livre.
- [Cote_Livre, "Restituer "] : Pour enregistrer la restitution de ce livre.

En plus de toutes ces requêtes, on trouve une requête d'authentification, car l'accès à l'application du lecteur est limité aux personnes autorisées. Cette requête a la forme : **[Nom_Utilisateur, Mot_De_Passe, " Authentification "]**.

Le lecteur doit permettre aux contrôleurs d'exploiter (traiter) les services (réponses) qui lui sont offerts par le serveur d'application, afin qu'ils réalisent leurs tâches (contrôler l'accès, gestion de bibliothèque).

La transmission et la réception des données entre le lecteur et le serveur d'application se font à base de sockets en mode TCP.

Le fonctionnement du lecteur est résumé par l'algorithme suivant :

Algorithme : Fonctionnement_Lecteur ;

// Initialisation.

Authentifier ← **Faux** ;

Sortir ← **faux** ;

Matricule_Etudiant ← " " ;

Cote_Livre ← " " ;

Application ← " " ;

Connexion_Serveur() ; *// Connexion au serveur.*

// Authentification du contrôleur.

Tant que (Authentifier = **Faux**)

{

→ Afficher la page d'authentification.

 Envoyer_Requête([Nom_Utilisateur, Mot_De_Passe, "Authentification"]);

//Envoyer la requête d'authentification.

 Recevoir(Réponse) ; *// Recevoir la réponse de serveur.*

Si Réponse = "Authentifié " **Alors** Authentifier ← **Vraie** ;

} //fin tant que.

// Traitement des opérations.

Répéter

Fonction ← Choisir_Fonction() ; *// Choisir une fonction par le contrôleur.*

Si (Fonction = "Contrôle_Accès ") **Alors** *//Contrôle d'accès.*

{
Matricule_Etudiant ← Lecteur_Tag(Carte_Etudiant) ; *// Récupérer le matricule de la carte d'étudiant.*

Envoyer_Requête([Matricule_Etudiant, "Contrôle_Accès "]) ;

Recevoir(Réponse) ; *// Recevoir la réponse de serveur.*

→ **Le contrôleur décide d'accorder ou de refuser l'accès au campus.**

}

Si (Fonction = "Gestion_Bibliothèque ") **Alors** *//Gestion de bibliothèque.*

{
Matricule_Etudiant ← Lecteur_Tag(Carte_Etudiant) ; *// Récupérer le matricule de la carte d'étudiant.*

Envoyer_Requête([Matricule_Etudiant, "Accès_Bibliothèque"]) ;

Recevoir(Réponse) ; *// Recevoir la réponse de serveur.*

→ **Le contrôleur décide d'accorder ou de refuser l'accès aux services de la bibliothèque.**

Si (Accès_Accordé) **Alors**

{

Envoyer_Requête([Matricule_Etudiant, "Livre "]) ;

Recevoir(Réponse) ; *// Recevoir la réponse de serveur.*

→ **Le contrôleur choisit l'opération à effectuer (emprunt ou restitution).**

Si (Opération = "Emprunt") **Alors**

{

Cote_Livre ← Lecteur_Tag(Livre) ; *// Récupéré la cote de livre.*

Envoyer_Requête([Cote_Livre, "Emprunter "]) ; *// Enregistrer l'emprunt*

}

Si (Opération = "Retour") **Alors**

{

Cote_Livre ← Lecteur_Tag(Livre) ; *// Récupéré la cote de livre.*

Envoyer_Requête([Cote_Livre, "Restituer "]) ; *//Enregistrer la restitution.*

}

Si (Application = "Sortir_Application ") **Alors** Sortir ← **Vraie** ;

Jusqu'à (Sortir = **Vraie**) ;

Remarque : La fonction Lecteur_Tag(objet) permet de récupérer l'identifiant qui se trouve dans le tag.

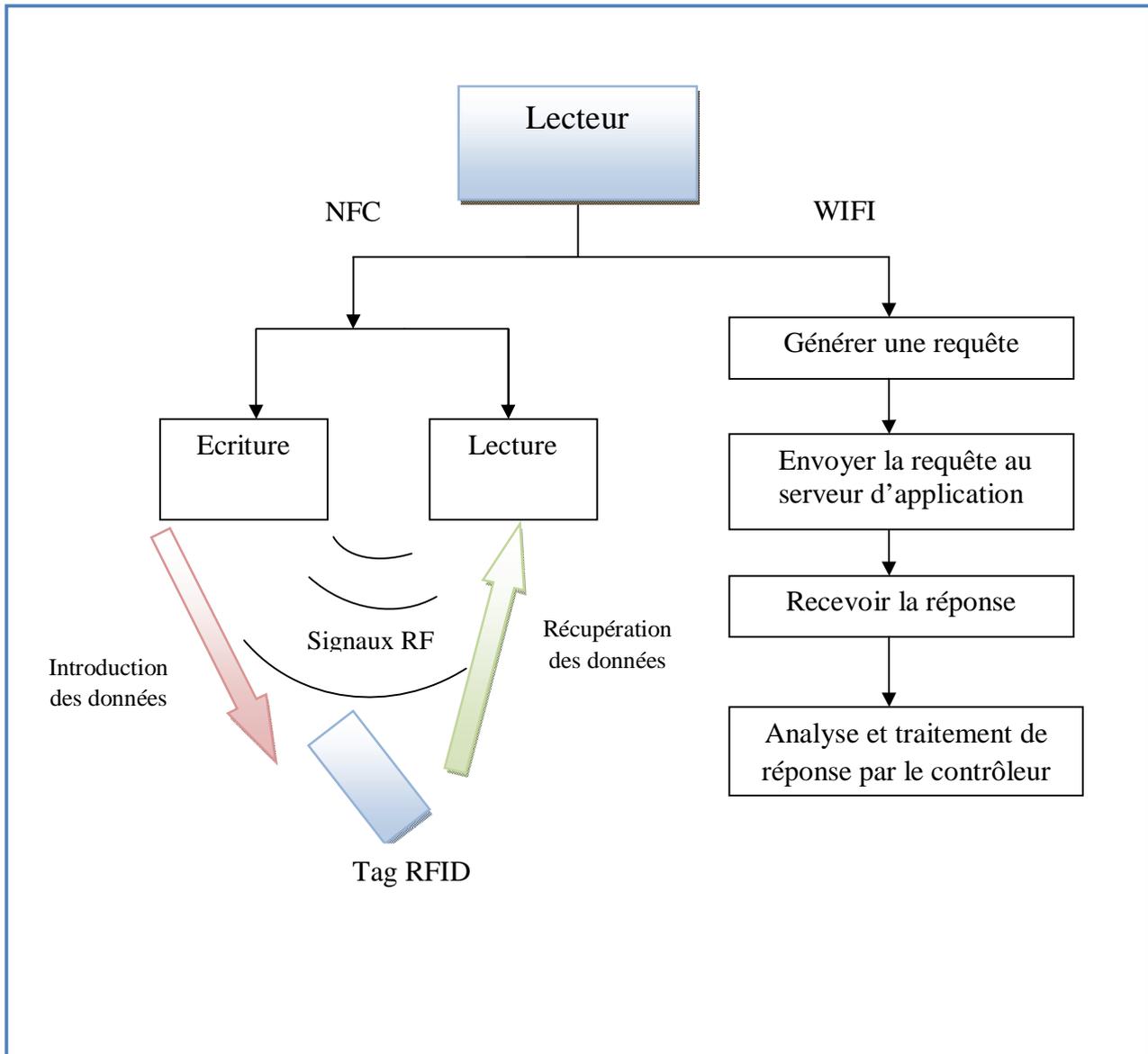


Figure III.2: Fonctionnement du lecteur.

III.3.3 Le contrôleur

Les tâches effectuées par le contrôleur (agent ou bibliothécaire) dépendent de l'opération demandée:

1. Cas de contrôle d'accès (Agent) :

- Il accorde ou refuse l'accès d'un étudiant au campus, après qu'il passe sa carte près du lecteur (10 cm au maximum).

2. Cas de gestion de bibliothèque (Bibliothécaire) :

- Il accorde ou refuse l'accès d'un étudiant aux services de la bibliothèque, après qu'il passe sa carte près du lecteur.
- Il vérifie la liste des livres empruntés et restitués, afin d'effectuer l'opération (emprunt ou retour), par exemple : une opération de restitution : si après la vérification de cette liste, il trouve un livre emprunté mais pas encore restitué.
- Il enregistre l'emprunt ou le retour d'un livre, après qu'il passe ce livre près du lecteur.

Remarque : Le contrôleur doit d'abord configurer la connexion au serveur et s'authentifier, avant qu'il accède aux applications proposées par le lecteur.

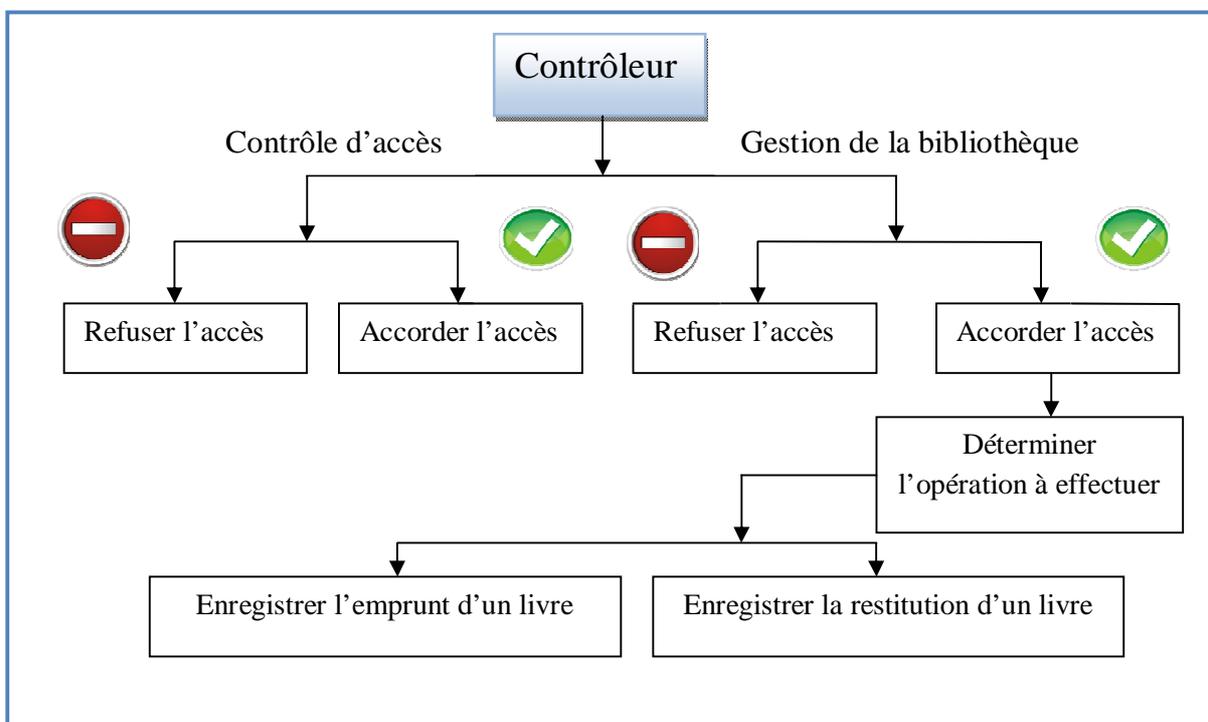


Figure III.3: Les tâches effectuées par le contrôleur.

III.3.4 Le serveur d'application :

Dans notre système le serveur d'application répond aux requêtes du client mobile (Smartphones), afin d'offrir les services (informations) demandés, mais faisant appel au serveur de base de données.

Le programme serveur possède deux modules : le premier, considéré comme étant le serveur maître, se met en écoute sur un port public dans une boucle infinie ; à l'arrivée d'une nouvelle connexion, il déclenche le deuxième module considéré comme le serveur esclave sous forme d'un thread, pour la suite de la communication et ainsi de suite.

Le serveur esclave se charge de :

1. Analyse et traitement des requêtes :

Le serveur esclave attend l'arrivée des requêtes client, lorsqu'une requête arrive, il l'analyse, puis lance le traitement associé.

La phase d'analyse consiste à déterminer depuis la requête reçue le service à lancer. Comme nous l'avons vu, chaque requête envoyée par le lecteur contient le code de service. Exemple : [Matricule_Etudiant, "Contrôle_Accès "] pour le contrôle d'accès au campus.

Après la spécification du service à lancer, le serveur d'application lance une requête SQL vers le serveur de base de données pour récupérer les informations nécessaires au service, par exemple : pour que le serveur d'application puisse répondre aux requêtes des clients [Matricule_Etudiant, "Contrôle_Accès "] il va lancer la requête : **Select * From Etudiant where Matricule =Matricule_Etudiant ;** tel que **Etudiant** est une table qui contient les informations des étudiants du campus.

2. Envoyer les réponses au lecteur :

Après l'exécution de la requête SQL par le serveur de base de données, il envoie les résultats au serveur d'application qui à son tour les envoie au lecteur. Les différentes informations envoyées par le serveur d'application vers le lecteur sont :

- Les **données de l'étudiant (nom, prénom, filière.....)** pour répondre aux requêtes: [Matricule_Etudiant, "Contrôle_Accès "], [Matricule_Etudiant, "Accès_Bibliothèque"]. Si l'étudiant correspond à cet matricule n'existe pas dans la base de données alors la réponse envoyée sera : **No**.

- La liste de tous les livres empruntés et restitués sous forme :

Cote de livre	Date de prêt	Date de restitution
---------------	--------------	---------------------

Pour répondre à la requête [Matricule_Etudiant, "Livre "].

- **Authentifié ou Non_Authentifié** pour répondre à la requête d'authentification.

Le fonctionnement du serveur d'application est résumé par l'algorithme suivant :

Algorithme : Fonctionnement du serveur d'application ;

// Serveur maitre ;

Tant que (vraie)

{

 Accepter la connexion d'un nouveau client ;

 Lancer Serveur_Esclave() ;

}

// Serveur esclave

Serveur_Esclave ()

{

 Quitter ← **faux** ;

Répéter

 Recevoir (Requête_Client) ;

 Service ← Analyser (Requête_Client) ; *// Déterminer le service à lancer.*

 Réponse ← Exécution(Service) ; *// Exécution de la requête SQL correspondante.*

 Envoyer (Client, Réponse) ; *// Envoyer la réponse.*

Jusqu'à (Quitter = **vraie**)

}

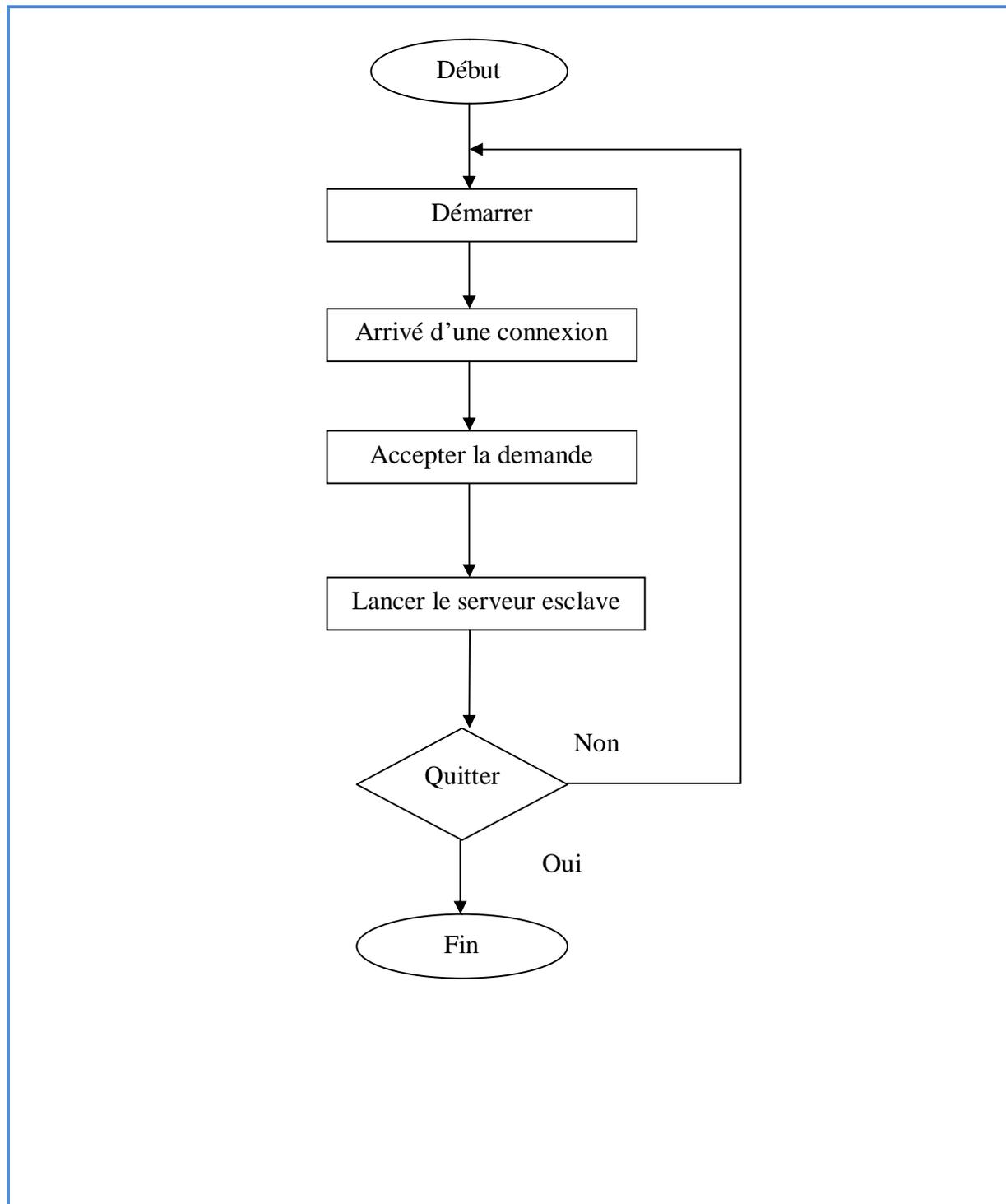


Figure III.4: Fonctionnement du serveur maitre.

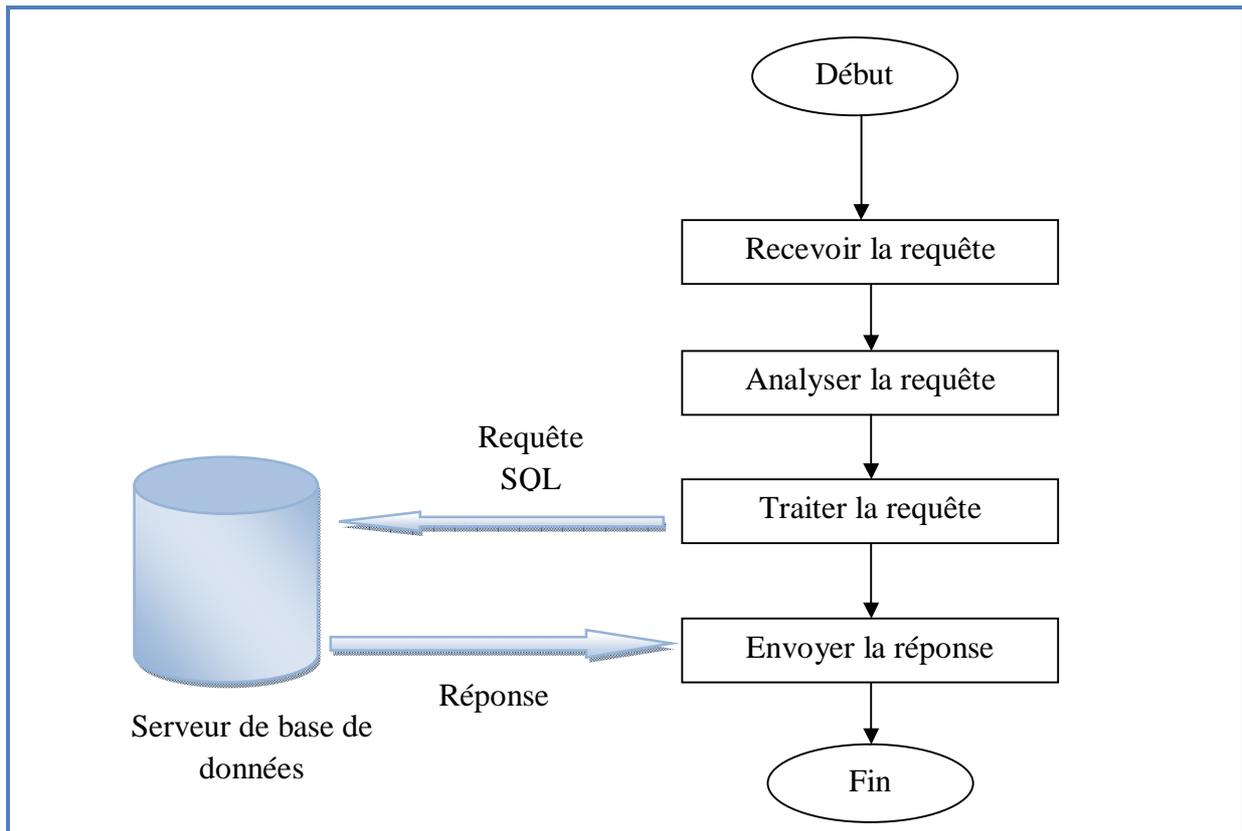


Figure III.5: Fonctionnement du serveur Esclave.

III.3.5 Le serveur de base de données :

Il stocke une base de données contenant les tables suivantes :

- 1) **Table Etudiant** : Elle stocke les données de tous les étudiant de campus, cette table est sollicitée par le serveur d'application lors du contrôle d'accès au campus.

Nom de colonne	Désignation	Type	Taille
<u>Matricule</u>	Matricule de l'étudiant	Caractère	20
Filiere	Filière de l'étudiant	Caractère	20
Inscrit	Niveau d'étude de l'étudiant	Caractère	40
Annee_Univ	Année universitaire	Caractère	10
Nom	Nom de l'étudiant	Caractère	20
Prenom	Prénom de l'étudiant	Caractère	20
Date_Ness	Date de naissance de l'étudiant	Date	/
Adresse	Adresse de l'étudiant	Caractère	40
Photo	Image de l'étudiant	Blob	/

Tableau III.1: Table Etudiant.

Comme chaque matricule est attribué à un seul étudiant, alors il est utilisé comme index de la table.

- 2) **Table Administrateur** : Elle contient les données des personnes qui ont accès à notre application.

Nom de colonne	Désignation	Type	Taille
<u>Login</u>	Login de l'Administrateur	Caractère	20
Mot_passe	Mot de passe de l'Administrateur	Caractère	20

Tableau III.2: Table Administrateur.

- 3) **Table Carte_Bibliotheque** : Elle contient les données des étudiants inscrits en bibliothèque. Elle est sollicitée par le serveur d'application afin de contrôler l'accès aux services de la bibliothèque

Nom de colonne	Désignation	Type	Taille
<u>Matricule</u>	Matricule de l'étudiant	Caractère	20
Filiere	Filière de l'étudiant	Caractère	20
Nom	Nom de l'étudiant	Caractère	20
Prenom	Prénom de l'étudiant	Caractère	20
Date_Ness	Date de naissance de l'étudiant	Date	/
Adresse	Adresse de l'étudiant	Caractère	40
Photo	Image de l'étudiant	Blob	/

Tableau III.3: Table Carte_Bibliothèque.

4) **Table Suivi_Bibliotheque** : Elle contient l'historique de l'étudiant dans la bibliothèque (liste de tous les livres empruntés et restitués), elle est sollicitée par le serveur d'application pour déterminer l'opération à effectuer (emprunt/restitution).

Nom de colonne	Désignation	Type	Taille
Matricule	Matricule de l'étudiant	Caractère	20
Date_pret	Date de prêt de livre	Date	/
Cote_livre	Cote de livre emprunté	Caractère	10
Date_Retour	Date de retour de livre	Date	/

Tableau III.4: Table Suivi_Bibliothèque.

III.4 Architecture de notre Application :

Le diagramme suivant décrit les différentes activités, classes et services constituant notre application:

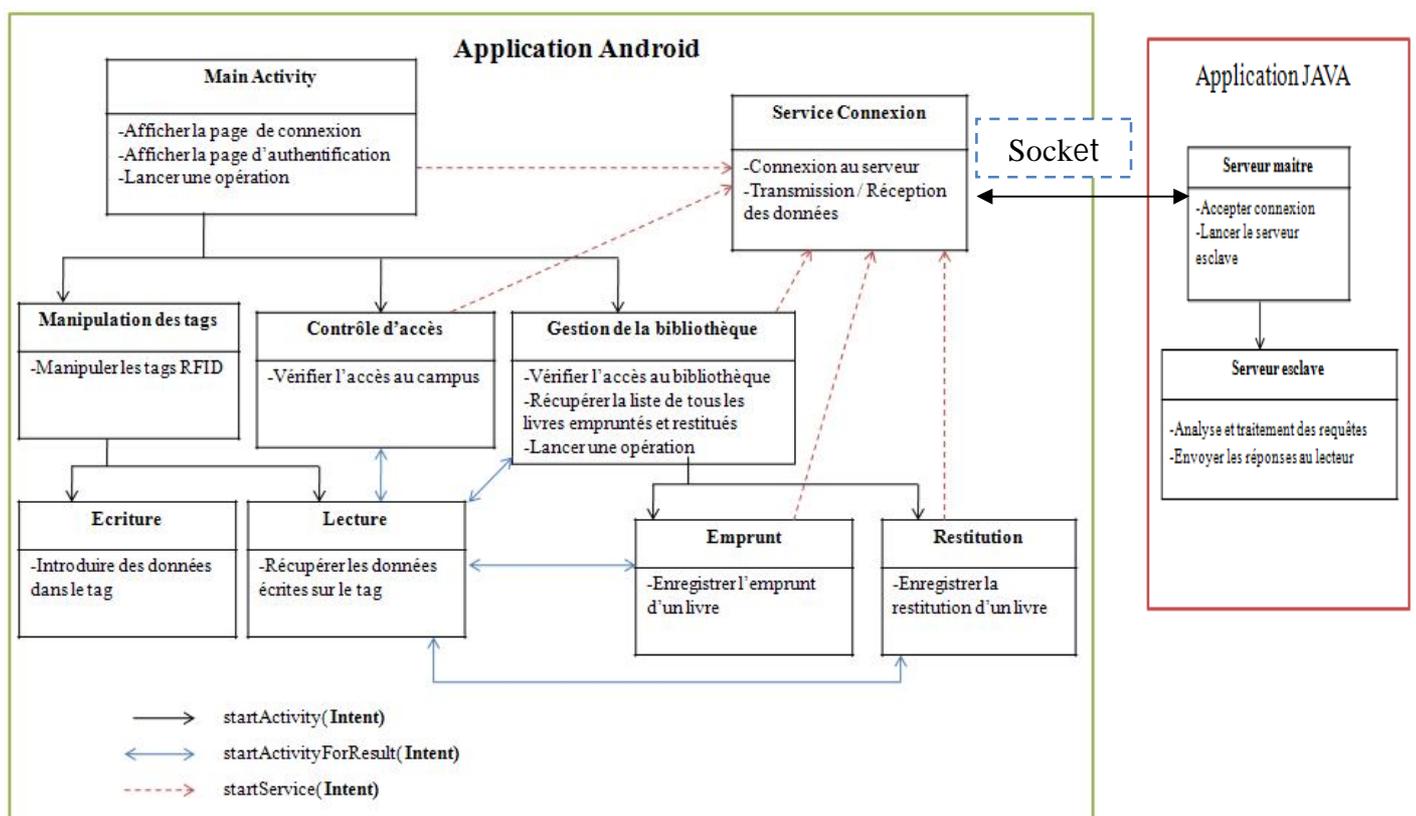


Figure III.6: Diagramme de notre application.

Voici une brève description des rôles de chaque activité, service ou classe :

- **Main Activity:** représente l'activité principale qui gère les autres activités comme le contrôle d'accès, la gestion de la bibliothèque et la manipulation des tags RFID.
- **Service Connexion:** gère la connexion avec le serveur et il s'occupe de la transmission et de la réception des données vers/depuis le serveur.
- **Contrôle d'accès:** accorde ou refuse l'accès d'un étudiant au campus, après qu'elle récupère son matricule en utilisant l'activité **lecture**.
- **Gestion de la bibliothèque:** accorde ou refuse l'accès d'un étudiant aux services de la bibliothèque, après la récupération de son matricule par l'activité **lecture**, elle vérifie la liste des livres empruntés et restitués, afin d'effectuer l'opération (emprunt ou retour).
- **Emprunter:** enregistre l'emprunt d'un livre après la récupération de sa cote par l'activité **lecture**.
- **Restituer :** enregistre la restitution d'un livre après la récupération de sa cote par l'activité **lecture**.
- **Manipulation des tags RFID:** manipule les tags RFID en faisant appel aux activités **lecture** et **écriture**.
- **Lecture :** récupère les informations écrites dans le tag.
- **Ecriture:** introduit des données dans le tag.
- **Serveur maître:** reçoit la requête de demande de connexion et lance le serveur esclave.
- **Serveur esclave:** analyse et traite les requêtes reçues et elle envoie les réponses au lecteur.

Conclusion :

Dans ce chapitre nous avons en premier lieu présenté les objectifs de notre projet puis l'architecture de notre système et nous avons aussi expliqué chacun de ses composants et le fonctionnement de notre application d'une manière détaillée. Le chapitre suivant sera consacré à présenter les outils matériels et logiciels qui sont utilisés pour le développement de notre application ainsi que la façon de son implémentation.

Introduction

Dans ce chapitre nous allons présenter les outils matériels et logiciels qui sont utilisés pour le développement de notre application ainsi que la façon de son implémentation, par la suite nous allons présenter les principales interfaces de notre application.

IV.1 Environnements et outils de développement :

IV.1.1 Partie matériel :

Elle est constituée d'un ensemble de ressources destinées à la mise en marche de notre application. Ainsi, nous citons :

IV.1.1.1 Tags RFID :

Pour identifier les étudiants et les livres au niveau de lecteur, nous avons utilisé des étiquettes RFID fournis par **3M** qui ont les caractéristiques suivantes :

- Type : passive (elles sont télé-alimentées par le lecteur).
- Type d'utilisation : elles sont des étiquettes à lecture et écriture multiple.
- Capacité mémoire : 256 bits.
- Elles respectent les normes ISO 15693 et 18000-3 mode 1.

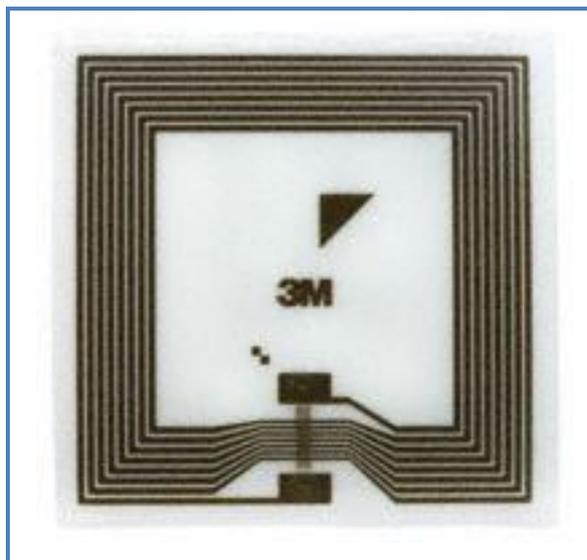


Figure IV.1: Tag RFID 3M.

IV.1.1.2 Smartphone:

Pour interroger les tags RFID, notre application Client est exécutée sur un Sony Ericson Xperia qui a les caractéristiques suivantes :

- Un processeur 1 GHZ et une RAM de 1Go.
- Portant le système Android (4.2.1) comme système d'exploitation.
- Compatible avec la technologie NFC.



Figure IV.2: Sony Ericson Xperia.

IV.1.1.3 PC portable:

Notre programme serveur de notre application est exécuté sur un PC portable qui a les caractéristiques suivantes :

- Processeur : Intel® core(TM) 2 duo CPU T2450 @ 2.00GHz 2.00 GHZ
- Mémoire vive : 2.00 Go
- Système d'exploitation : Windows XP Sweet 6.2.
- Disque dure : 120Go
- Eclipse INDIGO
- MySQL (version 5.1.36).

IV.1.2 Partie logiciel :

IV.1.2.1 Le langage JAVA : [18]

C'est le langage dans lequel sera écrit le programme Serveur d'application de notre application, nous ferons un tour d'horizon de ce puissant langage tout en restant le plus près possible de notre sujet, c.à.d. nous mettrons l'accent sur les mécanismes que nous allons utiliser tel que les sockets, les interactions avec les SGBD, notamment MYSQL qui fera aussi l'objet d'une autre étude dans ce qui suit.

➤ **Description :**

Java fait son apparition officielle dans la communauté Internet le 3 mai 1995 par la société Sun Microsystems. C'est un langage de programmation à usage général, évolué et orienté objet de très haut niveau capable de s'exécuter sur n'importe quelle machine, java est capable de tourner tout aussi bien sur un PC que sur un téléphone ou encore sur une carte à puce, ceci vient du fait que **Java**, à la différence de C++, est compilé en pseudo code (bytecode) et interprété par la **JVM** (Java Virtual Machine) afin de faire le lien entre l'environnement de travail et de ce dernier.

Parmi les caractéristiques les plus intéressantes de Java, on peut citer :

- Un langage orienté objet.
- Indépendance vis-à-vis de la plate forme.
- disposant de la capacité d'exécuter du code source extérieur de façon sécurisée permettant de compiler le code Java nativement, c'est-à-dire de produire un exécutable capable de fonctionner hors de l'environnement de Java.
- La programmation peut se faire pour des exemples simples avec le compilateur Javac, mais pour avoir plus de confort il est préférable d'utiliser un environnement de développement intégré ou IDE, celui que nous avons utilisé est l'environnement Eclipse.

➤ **Accès aux bases de données :**

Java permet aux programmeurs d'écrire du code qui met en œuvre les requêtes SQL pour retrouver des renseignements dans des bases de données relationnelles, mais ce n'est pas tout. Le langage Java est un langage universel qui, par essence, peut fonctionner sur différentes plates-formes. De plus, il faut que le programme puisse se connecter avec

n'importe quel serveur de base de données utilisée dans le commerce. Java dispose donc d'un système de connexion polyvalent.

La technologie de connectivité de base de données Java, appelé JDBC (Java Database Connectivity), est une bibliothèque de classe (et surtout d'interfaces) qui permet de travailler avec différents serveurs de bases de données relationnelles. Une fois que la connexion est établie, les interfaces prévues par la JDBC permettent d'obtenir tous les renseignements nécessaires en utilisant tout simplement le langage de requête SQL.

Pour se connecter à une base de données sous MYSQL, l'application Serveur doit charger un pilote qui n'est rien d'autre qu'une bibliothèque de classe. Le schéma suivant représente le mécanisme de connexion à des bases de données différentes par l'intermédiaire de pilotes JDBC.

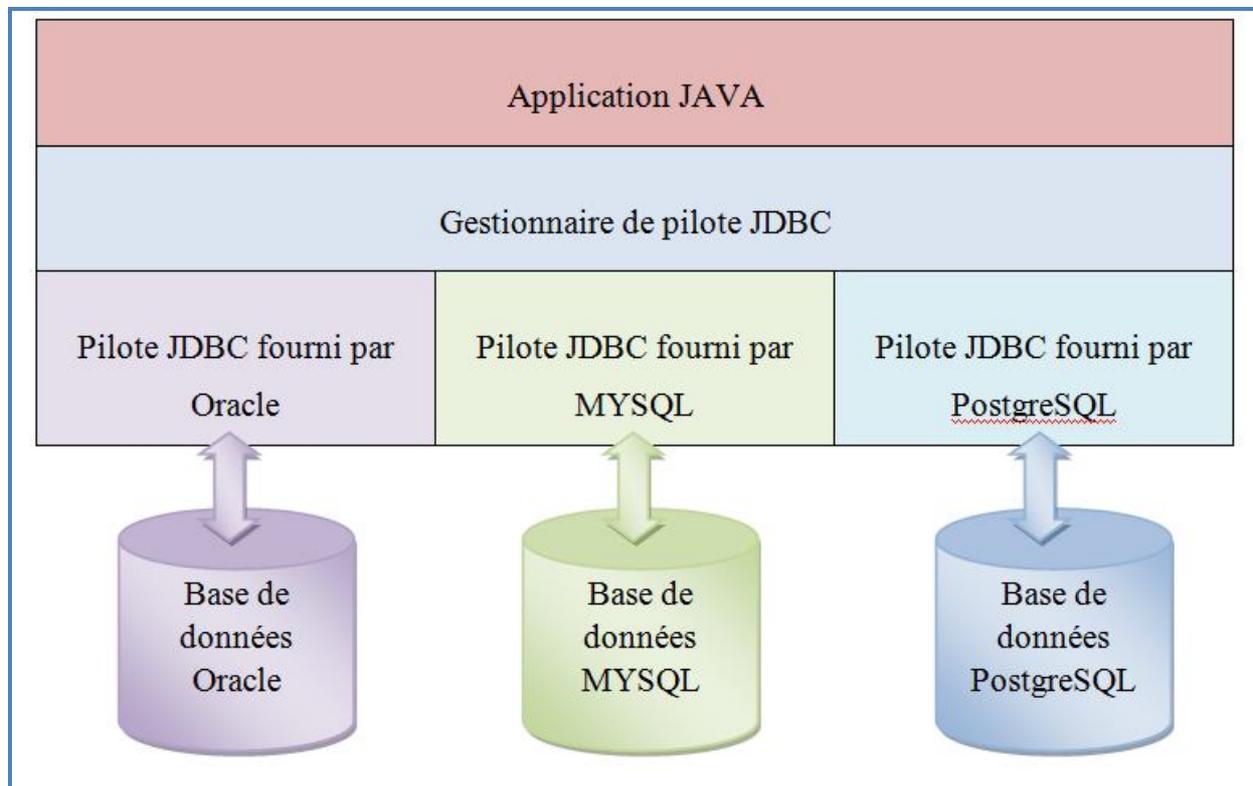


Figure IV.3: Accès aux bases de données.

Les classes de JDBC sont regroupées dans le package *java.sql* et sont incluses dans le JDK (Java Development Kit) à partir de sa version 1.1. La version 2.0 de cette API est incluse dans la version 1.2 du JDK.

Il y a 4 classes importantes, chacune correspond à une étape de l'accès aux données :

DriverManager : Charge et configure le pilote de la base de données.

Connexion : Réalise la connexion et l'authentification à la base de données.

Statement (et Prepared Statement): Contient la requête SQL et la transmet à la base de données.

ResultSet: Permet de parcourir les informations retournées par la base de données dans le cas d'une sélection de données.

➤ **Socket :**

La manière la plus ancienne de communiquer entre applications réparties a été l'utilisation des sockets. Les sockets sont des ports de communication ouverts au niveau des couches de réseau du système d'exploitation, et permettant de faire passer des flux (*streams*) d'octets.

Les Sockets permettent de communiquer en point à point en mode client/serveur. La socket serveur va accepter des connexions sur un port de communication donné, puis va analyser le flux obtenu ou écrire dans le flux. L'analyse du flux doit être faite manuellement. Le client va pour sa part se connecter et faire de même.

Les sockets existent sur les machines virtuelles Java sous deux formes : la classe *Socket* pour les applications clientes et la classe *ServerSocket* pour les applications serveur.

➤ **IDE Eclipse: [19]**

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

I.B.M. est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le coeur de son outil Websphere

Studio Workbench (WSW), lui même à la base de la famille des derniers outils de développement en Java d'I.B.M. Tout le code d'Eclipse a été donné à la communauté par I.B.M afin de poursuivre son développement.

Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

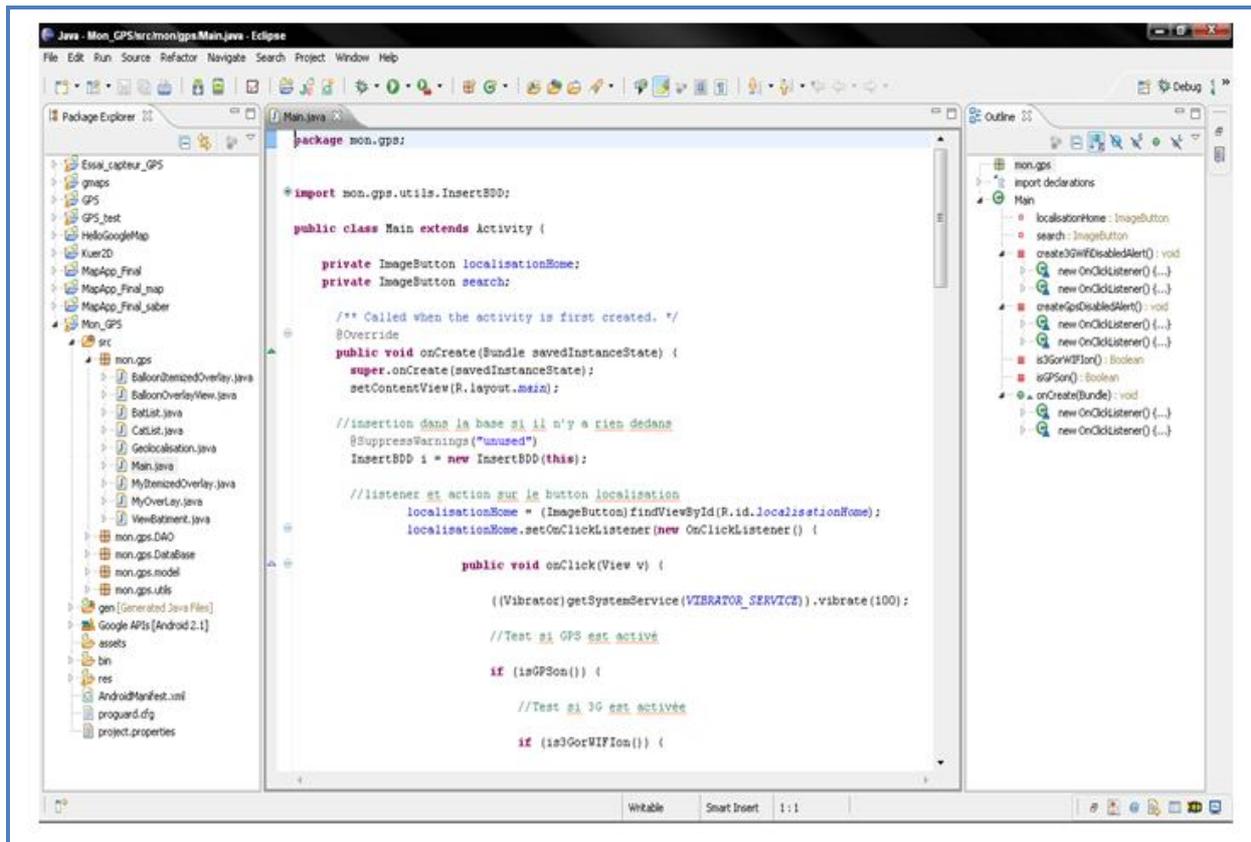


Figure IV.4: Interface d'Eclipse.

Dans le cadre de notre projet, nous avons utilisé la version Eclipse INDIGO.

IV.1.2.2 Prise en main de l'environnement Android :

La première étape de notre travail avec l'environnement Android a été d'appréhender le SDK, l'architecture et le développement d'une application ainsi que son déploiement sur un terminal Android.

➤ Le SDK d'Android: [20]

Un SDK, (**kit de développement**) est un ensemble d'outils que met à disposition un éditeur afin de nous permettre de développer des applications pour un environnement précis. Le SDK Android permet donc de développer des applications pour Android et uniquement pour Android. Le SDK d'Android contient un ensemble complet d'outils de développement. Il inclut un débogueur, des bibliothèques logicielles, un émulateur, de la documentation, des exemples de code et des tutoriaux. Les plateformes de développement prises en charge par ce kit sont les distributions sous Noyau Linux, Mac OS X 10.5.8 ou plus, Windows XP ou version ultérieure. L'IDE officiellement supporté est Eclipse combiné au plugin d'outils de développement d'Android (ADT).

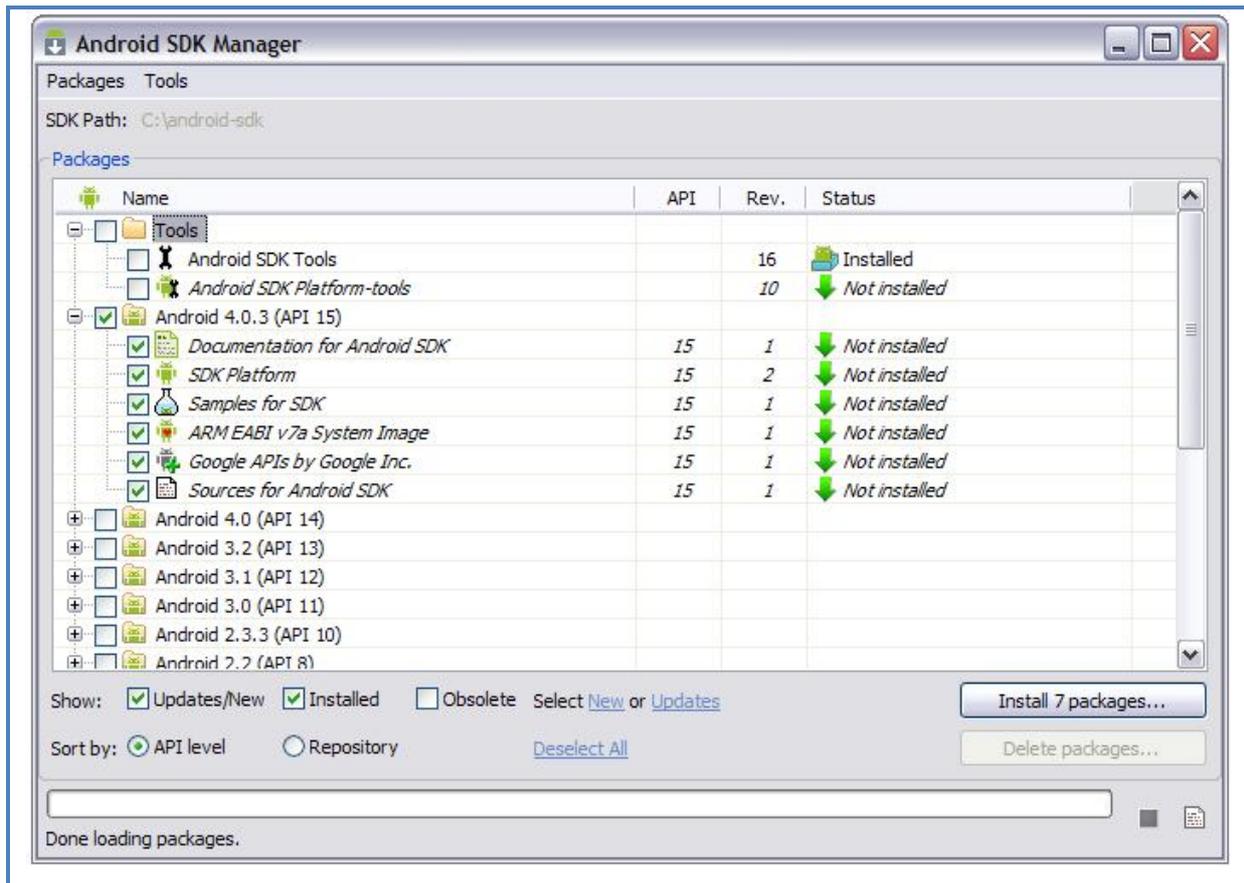


Figure IV.5: Android SDK Manager permet de choisir les paquets à télécharger.

Notre Application sur le client mobile (lecteur) a été développée pour la version 4.1.2 d'Android (API 16). Toutefois, nous avons choisit cette version car c'est la version la plus récente et elle permet de manipuler le capteur NFC de notre Smartphone.

➤ **ADT pour Android: [21]**

Pour développer sous Android, nous avons installé le plugin Android qui rajoutera à Eclipse les fonctionnalités spécialisées dans le développement sous Android. ADT (Android Development Tools) est un plugin pour l'IDE Eclipse est conçu pour donner un puissant environnement intégré pour la création d'applications Android. Il étend les fonctionnalités d'Eclipse pour permettre de mettre en place rapidement de nouveaux projets Android, créer une interface d'application, ajouter des packages basés sur l'API Android-cadre, débogage d'applications en utilisant les outils du SDK Android, et même exporter signé (ou non signé). Apk afin de distribuer une application.

➤ **L'émulateur de téléphone : Android Virtual Device [20]**

L'Android Virtual Device, aussi appelé AVD, est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui fait croire à votre ordinateur qu'il est un appareil sous Android. C'est la raison pour laquelle vous n'avez pas besoin d'un

périphérique sous Android pour développer et tester la plupart de vos applications. En effet, une application qui affiche un calendrier par exemple peut très bien se tester dans un émulateur, mais une application qui exploite le GPS, NFC ou Bluetooth doit être éprouvée sur le terrain (Smartphone physique) pour que l'on soit certain de son comportement.



Figure IV.6: Interface de l'émulateur Android.

IV.1.2.3 Serveur de base de données (SGBD MySQL): [22]

➤ Description

MySQL est un système de gestion de base de données. Son rôle est de stocker et de gérer une grande quantité de données en les organisant sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL, il est implémenté sur un mode client serveur avec du côté serveur : le serveur MySQL, et du côté client : les différents programmes et bibliothèques

La liste suivante décrit quelques fonctionnalités importantes de MySQL :

- ✓ **Multitraitement** : MySQL est multitraitement en utilisant les threads du noyau. Il peut utiliser plusieurs CPU.
- ✓ **Langages** : les applications de bases de données MySQL peuvent être écrites en C, C++, Eiffel, JAVA, PERL, PHP, PYTHON et TCL.
- ✓ **Multi plateformes** : prise en charge de plus de 20 plates-formes de système d'exploitation Windows, UNIX et LINUX.

- ✓ **Table** : MySQL stocke chaque table sous forme de fichier distinct dans le répertoire de bases de données. Le mélange des tables de différentes bases est supporté dans une même requête.

Dans notre application nous avons utilisé MySQL comme un serveur de base de données qui stocke et gère les données des étudiants.

➤ **Interface PhpMyAdmin :**

PhpMyAdmin est une application web qui permet de gérer un serveur de bases de données MySQL. Dans un environnement multiutilisateur, cette interface écrite en PHP permet également de donner à un utilisateur un accès à ses propres bases de données. Les fonctions principales de PhpMyAdmin:

- ✓ Création de nouvelles bases de données ;
- ✓ Création/suppression/modification des tables ;
- ✓ L'édition, l'ajout et la suppression de champs ;
- ✓ L'exécution de commandes SQL et de requêtes ;
- ✓ Chargement de fichier dans des tables ;
- ✓ Gérer les privilèges des utilisateurs.

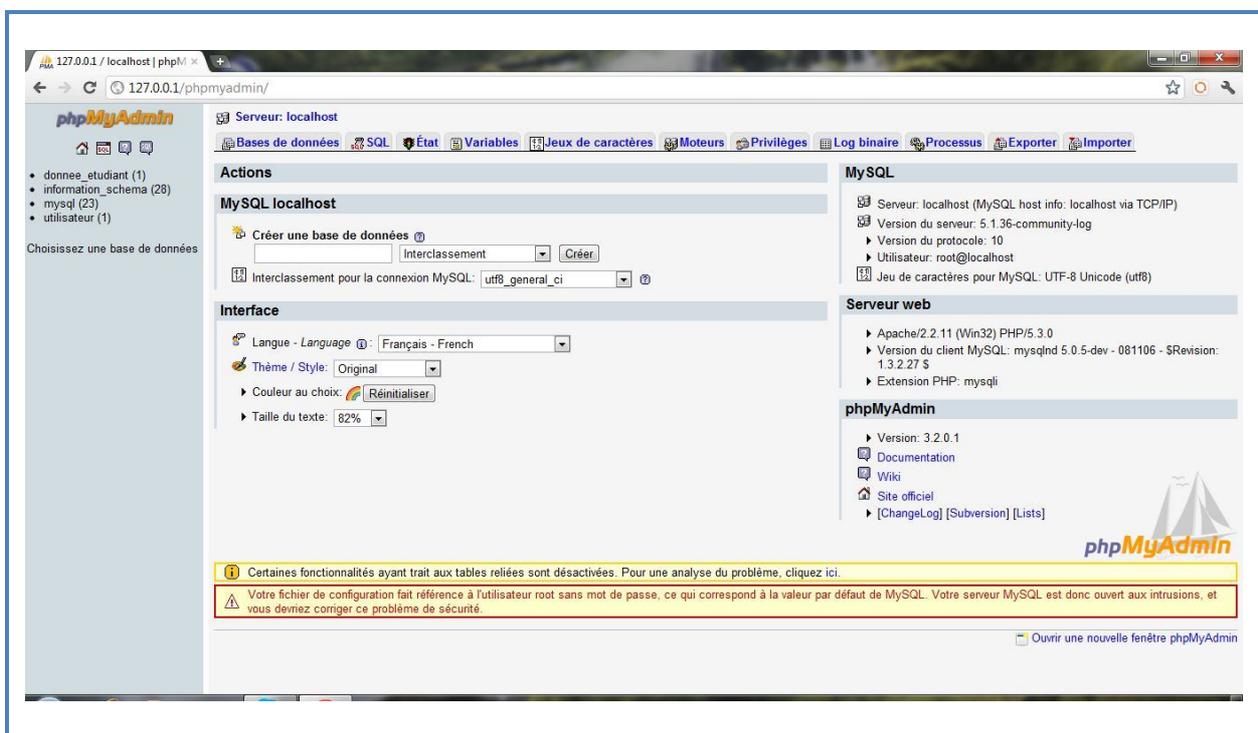


Figure IV.7: Interface de PhpMyAdmin.

IV.2 Présentation de l'application :

Notre système se décompose en trois parties :

La partie client, c'est une application Android qui s'exécute sur un Smartphone. Elle se connecte au serveur (PC) via WI-FI. La transmission et la réception des données se font à base de sockets en mode TCP/IP. La partie serveur de notre application, s'exécute sur le PC. Elle s'occupe de la connexion des clients, la transmission, la réception de données depuis ces derniers. Elle communique aussi avec le serveur de base de données (MySQL) pour insérer et récupérer des informations.

IV.2.1 La partie client :

Elle représente l'application Android, le déroulement de l'exécution se fait comme suit :

- Au lancement de l'application, la fenêtre de connexion demande au contrôleur de configurer la connexion au serveur par la spécification de l'adresse IP et le numéro de port.



Figure IV.8: La fenêtre de connexion.

- Une fois connecté, le contrôleur doit s'authentifier car l'accès à l'application est limité aux personnes autorisées.

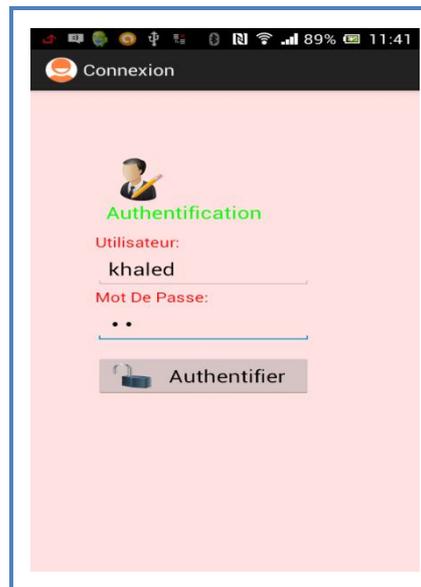


Figure IV.9: La fenêtre d'authentification.

- Le nom de l'utilisateur et le mot de passe vont être transmis au serveur et seront vérifiés auprès de la base de données. La réponse pour l'authentification sera transmise, si le nom d'utilisateur ou le mot de passe est invalide, un message d'erreur sera affiché. Sinon le contrôleur accède à la fenêtre «Services» de notre application.



Figure IV.10: Interface Services.

- Lors de l'apparition de la fenêtre « Services », le contrôleur peut accéder à chaque fonctionnalité de l'application grâce aux différents boutons présentés ci-dessous :
 1. Si le contrôleur a choisi « Contrôle d'accès », l'interface du lecteur de carte sera affichée :



Figure IV.11: Interface du lecteur de carte.

- A ce moment, le contrôleur va passer la carte de l'étudiant près du lecteur. Si l'étudiant existe dans la base de données alors ses informations vont être affichées, sinon une alerte va être lancée.



Figure IV.12: Interface d'affichage des informations de l'étudiant.

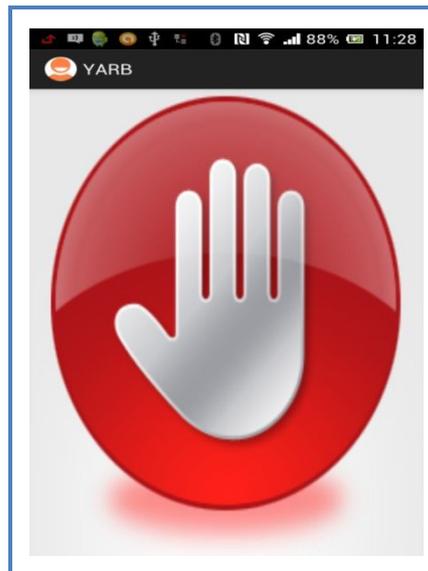


Figure IV.13: Interface d'affichage de message d'alerte.

2. Si le contrôleur a choisi « Gestion de la bibliothèque », l'interface du lecteur de carte sera affichée.



Figure IV.14: Interface de lecteur de carte.

- A ce moment le contrôleur va passer la carte de l'étudiant près du lecteur. Si l'étudiant est inscrit dans la bibliothèque alors ses informations vont être affichées, sinon une alerte va être lancée.



Figure IV.15: Interface d'affichage des informations de l'étudiant inscrit dans la bibliothèque.

- Lorsque le contrôleur va cliquer sur le bouton « livre », la liste de tous les livres empruntés et restitués par cet étudiant sera affichée :



Figure IV.16: La liste de tous les livres empruntés et restitués par l'étudiant.

- A partir de cette liste le contrôleur va déterminer l'opération à effectuer (emprunt / retour) :

- Si c'est une opération d'emprunt, le bouton « Restituer » sera désactivée, sinon si c'est une opération de restitution, le bouton « Emprunter » sera désactivée.



Figure IV.17: Emprunt d'un livre.



Figure IV.18: Restitution d'un livre.

- Une fois l'opération est choisie, le contrôleur va passer le livre afin d'enregistrer l'emprunt ou la restitution de ce dernier.



Figure IV.19: Interface de lecteur de livre.

3. Si le contrôleur a choisi « Manipulation des Tags », l'interface de lecture et d'écriture sera affichée.

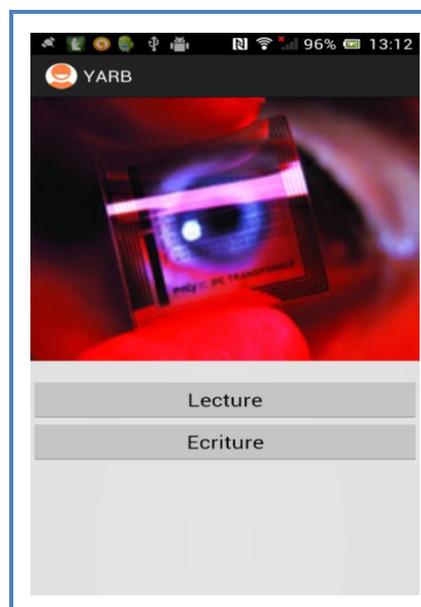


Figure IV.20: Interface de manipulation des Tags.

- Pour introduire des données dans les tags, le contrôleur appuie sur le bouton « Ecrire » afin de faire apparaître l'interface de l'écriture.

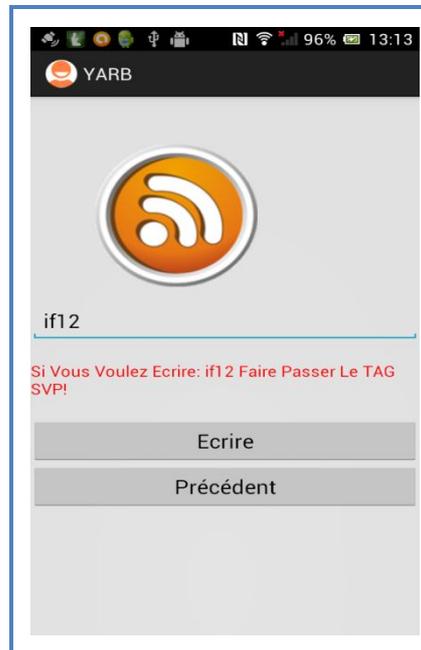


Figure IV.21: Interface d'écriture sur le tag.

- Pour vérifier les données écrites dans le tag, il va appuyer sur le bouton « Lecteur ».

IV.2.2 La partie Serveur d'Application:

Elle représente une application JAVA permettant de :

- I. Gérer les connexions des clients(Smartphones).
 - II. Gérer les données des étudiants (ajouter, supprimer ou modifier).
- Avant d'accéder à la page principale de l'application serveur l'administrateur doit s'authentifier.



Figure IV.22: Interface d'authentification.

- Après l'authentification, l'administrateur va accéder à la page principale de l'application serveur.



Figure IV.23: Interface principale de serveur.

Les figures ci-dessous représentent les interfaces des différentes tâches qui peuvent être effectuées par l'administrateur :

➤ **Lancer le serveur :**

Le bouton « Démarrer » permet à l'administrateur de lancer le serveur pour que les clients (Smartphones) puissent se connecter. Une fois le serveur est lancé, une nouvelle fenêtre apparaît contenant les informations de connexion, telles que l'adresse IP et le numéro de port du serveur, et elle contient aussi un bouton « Arrêter » permettant de mettre le serveur hors connexion.

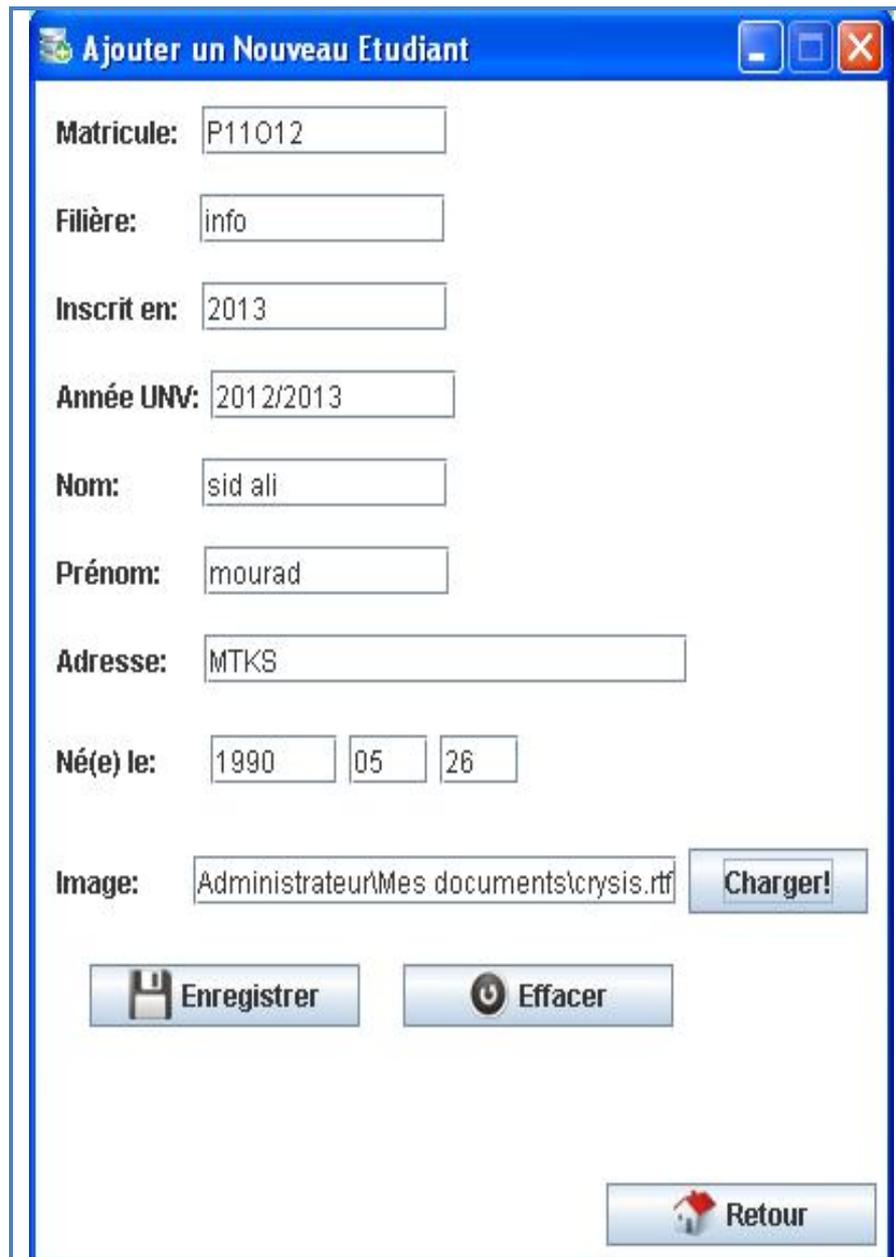


Figure IV.24: Interface qui indique que le serveur est lancé.

➤ **Gestions des étudiants :**

Cette partie permet de gérer les données de tous étudiants du campus.

- L'administrateur peut sélectionner le bouton « Ajouter », un formulaire d'ajout s'affiche. Il saisit les coordonnées de l'étudiant puis les valide en cliquant sur le bouton « Enregistrer ».



The screenshot shows a Windows-style window titled "Ajouter un Nouveau Etudiant". The window contains a form with the following fields and controls:

- Matricule:** Text box containing "P11012".
- Filière:** Text box containing "info".
- Inscrit en:** Text box containing "2013".
- Année UNV:** Text box containing "2012/2013".
- Nom:** Text box containing "sid ali".
- Prénom:** Text box containing "mourad".
- Adresse:** Text box containing "MTKS".
- Né(e) le:** Three separate text boxes for year, month, and day, containing "1990", "05", and "26" respectively.
- Image:** Text box containing the file path "Administrateur\Mes documents\crisis.rtf" and a "Charger!" button.
- At the bottom, there are three buttons: "Enregistrer" (with a floppy disk icon), "Effacer" (with a trash can icon), and "Retour" (with a home icon).

Figure IV.25: Interface d'ajout d'un nouvel étudiant.

- L'administrateur peut aussi sélectionner le bouton « Supprimer », un formulaire de suppression où il introduira le matricule de l'étudiant à supprimer sera affiché, une fois l'étudiant à supprimer est trouvé, il valide la suppression.



Supprimer un Etudiant

Matricule de l'étudiant SVP:

Matricule:

Filière:

Inscrit en:

Année UNV:

Nom:

Prénom:

Adresse:

Né(e) le:

Figure IV.26: Interface de suppression d'un étudiant.

- L'administrateur sélectionne le bouton « Modifier », puis remplit le formulaire de modification qui sera affiché en introduisant le matricule de l'étudiant à modifier, il saisit les nouvelles informations puis il les valide.

Modifier un Etudiant

Matricule de l'étudiant SVP:

Matricule:

Filière:

Inscrit en:

Année UNV:

Nom:

Prénom:

Adresse:

Né(e) le:

Image:

Figure IV.27: Interface de modification d'information d'un étudiant.

➤ **Gestion de la bibliothèque :**

On trouve les mêmes boutons présentées dans la partie précédente sauf que dans cette partie elles permettent de gérer les données des étudiants inscrits en bibliothèque.

IV.2.3 La partie serveur de base de données :

Elle contient une base de données MySQL contenant toutes les tables de notre application.

Conclusion :

Dans ce chapitre nous avons en premier lieu présenté l'environnement et les outils utilisés pour implémenter notre application. Puis nous avons présenté quelques interfaces du système réalisé.

Conclusion Générale

L'identification par radio fréquence RFID fait référence aux technologies qui utilisent les ondes radio pour identifier automatiquement des articles individuels ou groupés. La RFID promet de devenir la technologie de pointe dans l'identification automatique. De plus la RFID permet d'améliorer les services, réduire les coûts et réaliser des traitements professionnels comme la gestion des stocks, l'expédition, l'identification, et un suivi réellement efficace.

Le but de notre travail est d'étudier la technologie d'identification par radio fréquence RFID (Radio Frequency IDentification), à travers une application permettant le contrôle d'accès la gestion de l'emprunt et la restitution d'ouvrages dans une bibliothèque, une gestion d'une base de données à distance et assurer une communication avec des puces sans contact en utilisant le langage de programmation java Android, ce qui permet de gérer cette base de données facilement.

Les perspectives envisagées pour mettre en marche cette application sont :

- Commande de portique qui sont des portes qui s'ouvre automatiquement est sont doté de lecteur du puce RFID et qui remplacent l'agent contrôleur.
- Mise en place de solution pratique en utilisant des lecteur RFID dédiés.
- Mise en œuvre de nouvelle application comme la localisation de mobiles basée sur les puces RFID pouvant équiper les futurs documents comme les passeport, les cartes d'identités ,...etc.

Bibliographie

- [1] **Patrice Kadionik**, Les systèmes embarqués : une introduction, Décembre 2005.
- [3] **John Wiley & Sons**, Next Generation Mobile Communications Ecosystem: Technology Management for Mobile Communications, 25 Février 2011.
- [4] **John Wiley & Sons**, Developing Software for Symbian OS 2nd Edition: A Beginner's Guide to Creating Symbian OS v9 Smartphone Applications, 28 Février 2008.
- [8] **Frédéric Letient**, Etat de l'art et applications des RFID, Travail d'Etude et de Synthèse Technique en Electronique, Conservatoire National Des Arts ET Métiers Centre Régional de Grenoble, France, Juin 2008.
- [9] **Paulin I. Katamba**, Technologie RFID (radio frequency identification) concepts et stratégie de mise en œuvre, Mémoire présenté pour l'obtention du grade de maître es sciences (M. Se.), Université Laval, Canada, 2007.
- [10] **Anthony Ghioto**, Conception d'antenne de tags RFID UHF, Thèse de doctorat, Institut Polytechnique Grenoble, France, 2008.
- [11] **Oscar Botero**, Conception, analyse et évaluation d'un Réseau RFID Hétérogène, Thèse de doctorat, L'université Pierre & Marie Curie Université (Paris VI), France, 2012.
- [13] **Rami Khouri**, Modélisation comportementale en VHDL-AMS du lien RF pour la simulation et l'optimisation des systèmes RFID UHF et micro-ondes, Thèse de doctorat, Institut National Polytechnique de Grenoble, France.
- [14] **ALCOM Consulting et Newton Vaureal**, Etude sur les étiquettes électroniques et la traçabilité des objets, panorama stratégique, Ministère de l'Economie, des Finances et l'Industrie, France, 2007
- [16] **Aysegul Sarac**, Modélisation et aide à la décision pour l'introduction des technologies RFID dans les chaînes logistiques, Thèse de doctorat, Ecole nationale des Mines, Saint Etienne, France, 2010.
- [18] **Serge Tahé**, Apprentissage de langage JAVA, Université d'Angers, France, Septembre 98 - Révision juin 2002.
- [19] **Jean Michel Doudoux**, Développons en JAVA avec Eclipse, France, 2007.
- [22] **Jean Marie Defrance**, PHP MySQL avec Dreamweaver8, Eyrolles, 2006.

Webliographie

- [2] <http://wikipidia.org/>
- [5] <http://android.com>
- [6] <http://developer.android.com/reference/android/app/Activity.html>
- [7] www.hitechtools.com/RFID
- [12] <http://www.CNRFID.fr/la RFID/>
- [15] http://www-igm.univ-mlv.fr/~dr/XPOSE2007/mmadegar_rfid/
- [17] <http://sebrfini.developer.com/tutoriaux/android/nfc/>
- [20] www.siteduzero.com
- [21] <http://developer.android.com/tools/sdk/eclipse-adt.html>

Sommaire

Introduitin générale	1
----------------------------	---

Chapitre I: Généralités sur les système embarqués

Introduction	3
I.1 Définition d'un système embarqué.....	3
I.2 Historique.....	3
I.3 Composition d'un système embarqué.....	4
I.4 Caractéristiques des systèmes embarqués	4
I.5 Domaine d'application.....	6
I.6. Systèmes d'Exploitation Embarqués	6
I.6.1. Microsoft Windows Mobile.....	7
I.6.2 iOS	8
I.6.3 Palm OS	9
I.6.4 Symbian OS.....	9
I.6.5 BlackBerry OS.....	10
I.6.6 Linux OS	10
I.6.7 Bada	10
I.6.8 Android	10
a) Historique.....	11
b) Architecture du système d'exploitation Android.....	12
c) Applications Android.....	14
Conclusion.....	18

Chapitre II: Généralités sur les RFID

Introduction	19
II.1 Historique	19
II.2 Définition de la technologie RFID	20
II.3 Le principe de fonctionnement des systèmes RFID	20
II.4 Les composants d'un système RFID	21
II.4.1 Les lecteurs RFID (Interrogeurs).....	22
II.4.1.1 Les composants d'un lecteur RFID.....	22
II.4.1.2 Gammes de fréquences.....	23
II.4.2 Les tags RFID.....	24

II.4.2.1 Les composants d'un tag RFID	24
II.4.2.2 Types des tags RFID	25
II.4.2.3 Les différentes utilisations des tags RFID	27
II.4.3 Le middleware	28
II.5 Communication Tag passif Lecteur :	28
II.5.1 Nature de la communication	29
II.5.2 Codage des signaux	30
II.5.3 Modulation	30
II.5.4 Protocole de communication	31
II.6 Les fréquences d'un système RFID	32
II.7 La standardisation RFID	32
III.7.1 Les standards de données	33
II.7.2 Les standards de technologie	33
II.8 Applications RFID	34
II.9 Avantages et inconvénients de la RFID	35
II.9.1 Avantage	35
II.9.2 Inconvénients	36
II.10 La technologie NFC	37
II.10.1 Définition	37
II.10.2 Modes de fonctionnement du NFC	37
II.10.2.1 Mode émulation de carte	37
II.10.2.2 Mode lecteur	37
II.10.2.3 Mode pair-à-pair	38
II.10.3 NFC et Android	38
Conclusion	39
 Chapitre III: Conception de l'application	
Introduction	40
III.1 Présentation et objectifs du projet	40
III.1.1 Présentation	40
III.1.2 Objectifs :	41
III.2 Description du système	41
III.3 Description des composants du système	42
III.3.1 Les Tags RFID	42
III.3.2 Le Lecteur	43

III.3.3 Le contrôleur	47
III.3.4 Le serveur d'application	48
III.3.5 Le serveur de base de données	51
III.4 Architecture de notre Application	53
Conclusion.....	54
Chapitre IV: Réalisation de l'application	
Introduction	55
IV.1 Environnements et outils de développement	55
IV.1.1 Partie matériel	55
IV.1.1.1 Tags RFID.....	55
IV.1.1.2 Smartphone	56
IV.1.1.3 PC portable.....	56
IV.1.2 Partie logiciel	57
IV.1.2.1 Le langage JAVA	57
IV.1.2.2 Prise en main de l'environnement Android	60
IV.1.2.3 Serveur de base de données (SGBD MySQL)	62
IV.2 Présentation de l'application	64
IV.2.1 La partie client	64
IV.2.2 La partie Serveur d'Applicatio	71
IV.2.3 La partie serveur de base de données	77
Conclusion.....	77
Conclusion Générale.....	78
Bibliographie	79

Liste des figures

Chapitre I:

Figure I.1: Composition d'un système embarqué.

Figure I.2: Aperçu des versions d'Android.

Figure I.3: Architecture d'Android.

Figure I.4: Exemple d'une Activité d'une application Android.

Figure I.5: Cycle de vie d'une application Android.

Chapitre II:

Figure II.1: Fonctionnement d'un système RFID.

Figure II.2: Les composants d'un système RFID.

Figure II.3: Des lecteurs RFID.

Figure II.4: Les composants d'un lecteur RFID.

Figure II.5: Des tags RFID.

Figure II.6: Les composants d'une puce RFID.

Figure II.7: Eléments constitutifs du tag RFID.

Figure II.8: Les fonctions du middleware RFID.

Figure II.9: Représentation schématique d'une communication RFID.

Figure II.10: Exemple d'un EPC-96 bits.

Figure II.11: Les modes de fonctionnement du NFC.

Chapitre III:

Figure III.1: Architecture générale de notre système.

Figure III.2: Fonctionnement de lecteur.

Figure III.3: Les tâches effectuées par le contrôleur.

Figure III.4: Fonctionnement du serveur maitre.

Figure III.5: Fonctionnement du serveur Esclave.

Figure III.6: Diagramme de notre applicatioin

Chapitre IV:

Figure IV.1: Tag RFID 3M.

Figure IV.2: Sony Ericson Xperia.

Figure IV.3: Accès aux bases de données.

Figure IV.4: Interface d'Eclipse.

Figure IV.5: Android SDK Manager permet de choisir les paquets à télécharger.

Figure IV.6: Interface de l'émulateur Android.

Figure IV.7: Interface de PhpMyAdmin.

Figure IV.8: La fenêtre de connexion.

Figure IV.9: La fenêtre d'authentification.

Figure IV.10: Interface Services.

Figure IV.11: Interface de lecteur de carte.

Figure IV.12: Interface d'affichage des informations de l'étudiant.

Figure IV.13: Interface d'affichage de message d'alerte.

Figure IV.14: Interface de lecteur de carte.

Figure IV.15: Interface d'affichage des informations de l'étudiant inscrit dans la bibliothèque.

Figure IV.16: La liste de tous les livres empruntés et restitués par l'étudiant.

Figure IV.17: Emprunter un livre.

Figure IV.18: Restitution un livre.

Figure IV.19: Interface de lecteur de livre.

Figure IV.20: Interface de manipulation des tags.

Figure IV.21: Interface d'écriture sur le tag.

Figure IV.21: Interface d'authentification.

Figure IV.22: Interface principale de serveur.

Figure IV.23: Interface qui indique que le serveur est lancé.

Figure IV.24: Interface d'ajout d'un nouvel étudiant.

Figure IV.25: Interface de suppression d'un étudiant.

Figure IV.26: Interface de modification d'information d'un étudiant.

Liste des tableaux

Chapitre I:

Tableau I.1: Evolution des versions Microsoft Windows Mobile.

Tableau I.2: Evolution des versions de Palm OS.

Chapitre II:

Tableau II.1: Plage des fréquences d'un Système RFID.

Tableau II.2: Quelques standards ISO.

Tableau II.3: Exemples d'applications RFID.

Chapitre III:

Tableau III.1: Table Etudiant.

Tableau III.2: Table Administrateur.

Tableau III.3:Table Carte_Bibliothèque .

Tableau III.4:Table Suivi_Bibliothèque.

Acronyms

ADT	Android Devloppement Tools
AIDC	Automatic Identification Data Capture
API	Application Programming Interfaces
ASIC	Application Sepecific Integrated Circuit
ASSP	Application Sepecific Standard Parts
AVD	Andriod Virtuls Device
BF	Basse frequency
EPC	Electronic Product Code
FPGA	Field programmable Gate Array
FSK	frequency Shift Keying
GPS	Global Positioning Systems
HF	Haut Frequence
IFF	Identity Friend or Foe
ISO	International Organization for Standardization
JDBC	Java Data Base connectivity
JDK	Java Development Kit
JVM	Java Vertual Machine
NFC	Near Field Communication
NRZ	Non Retour à Zéro
OHA	Open Hanset Alliance
OOK	On Of Keying
PDA	Personal Digital Assistant

RF	Radio Frequencies
RFID	Radio Frequency Identification
RTF	Reader Talks First
RZ	Retour à Zéro
SDK	Software Development Kit
SHF	Super Frequency
TCP	Transmission Control Protocol
TTF	Tag Talks First
UCC	Uniform Code Council
UHF	Ultra Haut Frequene
WIFI	Wireless Fidelity
WORM	Write Once ,Read Many

Conclusion Générale

Bibliographie

Chapitre I

Généralités sur les systèmes embarqués

Chapitre II

Généralités sur les RFID

Chapitre III

Conception de l'application

Introduction Générale

Chapitre IV

Réalisation de l'application