

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université Dr Taher Moulay SAIDA**

**Faculté : technologie**

**Département : d'informatique**



**MEMOIRE DE MASTER**

*OPTION : Informatique*

Spécialité : Réseaux informatiques et systèmes Réparties

**THEME**

---

**Système de gestion de RNDV basé sur  
Android**

---

**Présenté par :**

MEDJBER Wassila

**encadré par :**

Dr. MEKKAOUI Kheireddine

**2020 /2021**

## REMERCIEMENT

*En témoignage de notre gratitude et de notre profond respect nous adressons nos vifs et sincères remerciements à :*

*À MON encadreur DR. Mekkaoui Kherddine pour ses conseils, sa disponibilité.*

*Les jurys pour leurs efforts et leur soin apporté à notre travail. Aux enseignants de notre université et département informatique. Nous ne pouvons nommer ici toutes les personnes qui de près ou de loin nous ont aidé et encouragé mais nous les en remercions vivement.*

*Enfin, nous tenons à dire combien le soutien quotidien de notre famille a été important tout au long de ces quelques années, nous leur devons beaucoup*

---

## Dédicace

*Je dédie ce modeste travail a mes parents*

*A mon frère et ma sœur*

*A tout ma famille qui mon aidée et encouragé*

*A tout le département d'informatique.*

# Table des matières :

REMERCIEMENT .....	II
Dédicace .....	III
Table des matières : .....	IV
Liste de figures : .....	VII
Introduction générale.....	VIII
I. Chapitre 1 : Réseaux sans fil.....	1
I.1 Introduction :.....	2
I.2 Définition : .....	3
I.2.1 Les catégories des réseaux sans fil : .....	3
I.2.1.1 Selon la zone de couverture : .....	4
• Le réseau personnel sans fil (WPAN) :.....	4
• Le réseau local sans fil (WLAN) :.....	4
• Le réseau métropolitain sans fil (WMAN) : .....	5
• Le réseau étendu sans fil (WWAN) :.....	5
I.2.1.2 Selon l'infrastructure : .....	5
• Réseaux cellulaires (avec infrastructure) : .....	6
• Réseaux ad hoc (sans infrastructure) : .....	6
• Définition d'Un réseau ad hoc:.....	7
I.3 Architecture client/serveur : .....	10
I.3.1 Principe du client/serveur .....	10
I.3.2 Communication client/serveur : .....	12
I.3.3 Avantages : .....	12
I.3.4 Inconvénients .....	12
I.4 Java Socket Programmation .....	13
I.4.1 classe de socket .....	14
I.4.1.1 Méthodes importantes.....	14
I.4.1.2 Classe ServerSocket .....	14
I.4.1.3 Méthodes importantes : .....	14
I.4.2 Exemple de programmation de socket Java : .....	14
I.5 Conclusion : .....	16

II.	Chapitre 2 : Android .....	17
II.1	Introduction :.....	18
II.2	Logo .....	18
II.3	Architecture d'Android.....	20
II.4	L'environnement Dalvik .....	22
II.5	Cycle de vie d'une activité (Activity Lifecycle).....	23
II.6	Conclusion: .....	25
III.	Chapitre3 : La conception.....	26
III.1	Introduction :.....	27
III.2	Présentation d'UML :.....	27
III.2.1	Types de diagrammes UML .....	28
III.2.1.1	Diagrammes UML structurels.....	28
III.2.1.2	Diagrammes UML comportementaux .....	29
III.3	La conception : .....	29
III.3.1	Diagramme de cas d'utilisation :.....	30
III.3.2	Diagramme de séquence :.....	31
III.3.2.1	L'inscription : .....	32
	.....	32
III.3.2.2	Connexion :.....	33
III.3.2.3	Réservation :.....	34
III.3.2.4	Diagramme de classe :.....	34
III.4	Conclusion : .....	35
IV.	Chapitre 4 :Implémentation .....	36
IV.1	Introduction:.....	37
IV.2	Environnements de développement :.....	37
IV.2.1	Netbeans : .....	37
IV.2.2	AndroidStudio :.....	37
IV.2.3	MYSQL : .....	38
IV.3	Langage de programmation : Java .....	38
IV.4	Description de l'application :.....	38
IV.4.1	Coté serveur : .....	39
IV.4.2	Coté Client : .....	40
IV.4.3	Les tables de base de données :.....	43
IV.5	Conclusion : .....	44

---

Conclusion Générale :.....	45
Bibliographie.....	46

## Liste de figures :

Figure 1 Les catégories des réseaux sans fil .....	4
Figure 2 des réseaux mobiles avec infrastructure .....	6
Figure 3 Réseau en mode ad hoc .....	7
Figure 4 Le changement de la topologie des réseaux ad hoc .....	8
Figure 5 architecture client/serveur et architecture p2p.....	11
Figure 6 Socket API .....	13
Figure 7 logo android 2007 /2014 .....	19
Figure 8 logo android 2014/2019 .....	19
Figure 9 Logo android 2019 .....	19
Figure 10 Architecture d'Android.....	20
Figure 11 compilation et à l'exécution d'un programme Java standard .....	21
Figure 12 la compilation et à l'exécution d'un programme Android standard.....	22
Figure 13 cycle de vie d'une activité android .....	24
Figure 14 Diagramme de cas d'utilisation .....	30
Figure 15 diagramme de séquence de l'inscription .....	32
Figure 16 diagramme de séquence de Connexion .....	33
Figure 17 diagramme de séquence de Réservation .....	34
Figure 18 diagramme de Classe.....	35
Figure 19 coté serveur.....	39
Figure 20 Start Server .....	39
Figure 21 la table patient .....	40
Figure 22 connecter avec serveur .....	40
Figure 23 Connexion.....	41
Figure 24 inscription.....	42
Figure 25 réserver un rendez vous.....	43

---

# Introduction générale

Il ne fait désormais plus aucun doute que l'informatique est la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité moderne. En effet, les logiciels informatiques proposent maintenant des solutions à tous les problèmes de la vie, aussi bien dans des domaines professionnels que pour des applications personnelles. Leurs méthodes de conception et de développement ont vu l'évènement d'autant de technologies qui facilitent leur mise en place et leurs donnent des possibilités et des fonctionnalités de plus en plus étendues, et surtout avec l'interaction d'internet qui est l'outil par excellence pour se faire connaître de tous .C'est aussi le meilleur moyen pour les organismes ou les entreprises de vendre leurs produits et services. Les particuliers, les grandes structures commerciales, les sociétés aussi bien privées que publics bref tous les secteurs pour se faire connaître ou mieux servir leur clientèle utilisent les médias (radio, télévision, ...). Mais tous ces différents moyens précédemment cités ont une portée faible (rayon d'action très limités) et ne donnent pas la possibilité aux usagers de pouvoir interagir avec l'organisation. De plus Internet aujourd'hui occupe une place de choix dans la vie de tous les jours à travers ses multiples services et plus particulièrement son gain de temps. On a tenu compte de cette innovation et de tous ces opportunités qu'elles présentent tel que la création d'un site web dynamique qu'on l'en a opté pour notre projet de fin d'étude.

Ainsi, l'objectif de notre projet est de réaliser une application informatique une application mobile facile à s'intégrer dans l'environnement de travail de multiples secteurs nous avons choisi un exemple du secteur médical. Cette application vise essentiellement à diminuer la charge quotidienne des gens en diminuant le temps d'attente des clients dans la prise d'un rendez-vous.

## Plan du document

Le document est organisé en 4 chapitres. Le découpage a été fait de manière à couvrir les domaines impliqués dans notre travail.

**Le chapitre 1** : est consacré à la Définition Les Réseau sans fil et ses catégorie, en plus l'architecture Client/Server et une petite définition de Socket java.

**Le chapitre 2** : est présenté Le Système Android.

**Le chapitre 3** : concernant la partie conception de notre application gestion des rendez vous je choisi UML qui est destiné à faciliter la conception des

---

documents nécessaires au développement d'un logiciel orienté objet, comme standard de modélisation de l'architecture logicielle. et on définit les diagrammes de séquence, classe et de cas d'utilisation.

**Le chapitre 4 :** c'est la partie implémentation de notre application android, nous expliquerons le choix du langage de programmation, puis on a expliqué le fonctionnement et l'objectif de notre projet .

---

# **I. Chapitre 1 : Réseaux sans fil**

## I.1 Introduction :

On nomme réseau un ensemble d'acteurs, d'agents économiques, de nœuds, oubliés de communication grâce auxquels les messages circulent. L'information se concentre et se redistribue ainsi. On parle d'un réseau.

Un réseau informatique est un ensemble d'équipements reliés entre eux pour échanger des informations. Par analogie avec un filet (un réseau est un « petit rets », c'est-à dire un petit filet), on appelle nœuds (*node*) l'extrémité d'une connexion, qui peut être une intersection de plusieurs connexions (un ordinateur, un routeur, un concentrateur, un commutateur).

Les infrastructures ou supports peuvent être sur des câbles dans lesquels circulent des signaux électriques, l'atmosphère (ou le vide spatial) où circulent des ondes radio, ou des fibres optiques qui propagent des ondes lumineuses. Elles permettent de relier « physiquement » des équipements assurant l'interconnexion des moyens physiques qui sont définis par des protocoles. Les équipements d'un réseau sont connectés directement ou non entre eux, conformément à quelques organisations types connues sous le nom de topologie de réseau. Les contraintes liées à l'application exigent souvent l'utilisation de telle ou telle topologie.

Les protocoles de communication permettent de définir de façon standardisée la manière dont les informations sont échangées entre les équipements du réseau: il s'agit de procédures qui contrôlent le flux d'information entre deux équipements. Des logiciels spécifiques qui gèrent ces protocoles sont installés sur les équipements d'interconnexion comme les commutateurs réseau, les routeurs, les commutateurs téléphoniques, les antennes GSM, etc.

Les réseaux informatiques sont classés suivant leur portée :

- le réseau personnel (PAN) relie des appareils électroniques personnels ;
- le réseau local (LAN) relie les ordinateurs ou postes téléphoniques situés dans la même pièce ou dans le même bâtiment ;
- le réseau métropolitain (MAN) est un réseau à l'échelle d'une ville ;

- le réseau étendu (WAN) est un réseau à grande échelle qui relie plusieurs sites ou des ordinateurs du monde entier.

Et selon l'architecture :

- avec infrastructure.
- réseaux ad hoc.

## **I.2 Définition :**

Un réseau sans fil est un réseau informatique ou numérisé qui connecte différents postes ou systèmes entre eux par ondes radio. Il peut être associé à un réseau de télécommunications pour réaliser des interconnexions entre nœuds. La norme la plus utilisée actuellement pour les réseaux sans-fil est la norme IEEE802.11, mieux connue sous le nom de Wifi. Le rayonnement géographique des ondes est relativement limité étant donné la faible puissance d'émission des solutions matérielles actuelles. Pour cette raison, les réseaux sans-fil apparaissent comme des réseaux internes, propres à un bâtiment, soit comme réseau d'entreprise, soit comme réseau domestique. Néanmoins, des projets de réalisation de réseaux à grande échelle ont vu le jour, notamment le WiMAX. [1]

### **I.2.1 Les catégories des réseaux sans fil :**

Il existe plusieurs catégories de réseaux sans fil qui diffèrent par le périmètre géographique qu'ils couvrent ainsi que par les types d'applications supportées. Le schéma suivant illustre les catégories des réseaux sans fil. [1]

### ***1.2.1.1 Selon la zone de couverture :***

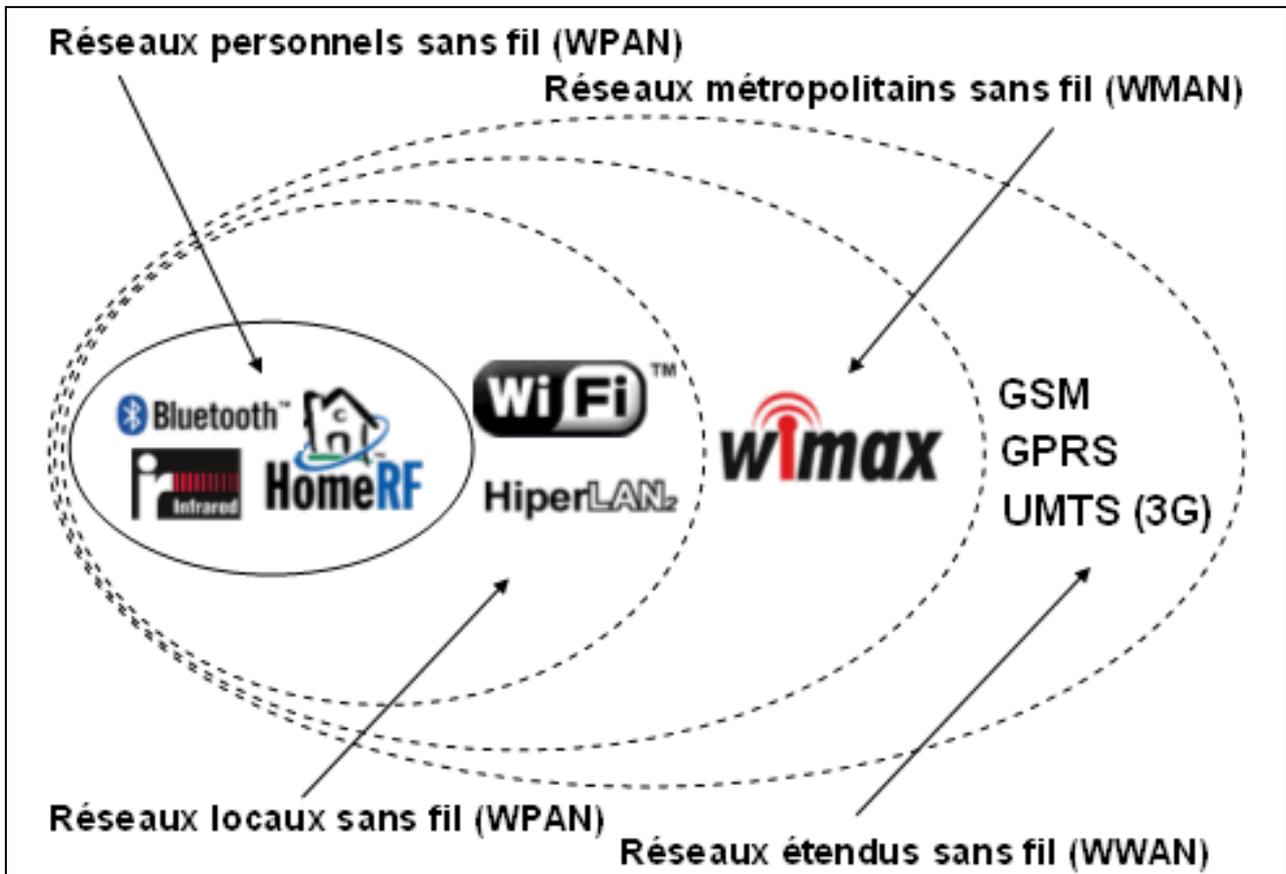


Figure 1 Les catégories des réseaux sans fil [16]

- ***Le réseau personnel sans fil (WPAN) :***

Il concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines de mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, PDA...). Il existe plusieurs technologies utilisées pour les WPAN : **La technologie Bluetooth, La technologie ZigBee, Les liaisons infrarouges.** [1]

- ***Le réseau local sans fil (WLAN) :***

C'est un réseau permettant de couvrir une portée d'environ une centaine de mètres. Il permet de relier entre-deux les terminaux présents dans la zone de couverture. Il existe deux technologies concurrentes :

- **Les réseaux Wi-Fi (Wireless-Fidelity) :**

Ils proviennent de la norme IEEE 802.11, qui définit une architecture cellulaire. On y trouve principalement deux types de réseaux sans fil : Ceux qui travaillent à la vitesse de 11 Mbits/s à 2.4 GHz (IEEE 802.11b) et ceux qui montent à 54 Mbits/s à 5 GHz (IEEE 802.11 a/g)

- **Les réseaux HiperLAN 2 (High Performance LAN 2.0) :**

Ils découlent de la norme Européenne élaborée par l'ETSI (Européen Télécommunications Standards Institute). HiperLAN2 permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5 150 et 5 300 MHz [01]. Ce type de réseau n'a pas reçu autant de succès que la technologie Wi-Fi.[1]

- ***Le réseau métropolitain sans fil (WMAN) :***

Il est connu aussi sous le nom de Boucle Locale Radio (BLR). Il convient de rappeler que la BLR permet, en plaçant une antenne parabolique sur le toit d'un bâtiment, de transmettre par voie hertzienne de la voix et des données à haut débit pour l'accès à l'internet et la téléphonie. Il existe plusieurs types de réseaux WMAN dont le plus connu est : **les réseaux Wimax [16].**

- ***Le réseau étendu sans fil (WWAN) :***

Il est connu sous le nom de réseau cellulaire mobile et il est le plus répandu de tous puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont les suivantes : GSM (Global System for Mobile Communication), GPRS (General Packet Radio Service), UMTS (Universal Mobile Télécommunication System) [01].

### ***1.2.1.2 Selon l'infrastructure :***

Les environnements mobiles sont des systèmes composés de sites mobiles et qui permettent à leurs utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. Les réseaux mobiles ou sans fil, peuvent être

classés en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure. [16].

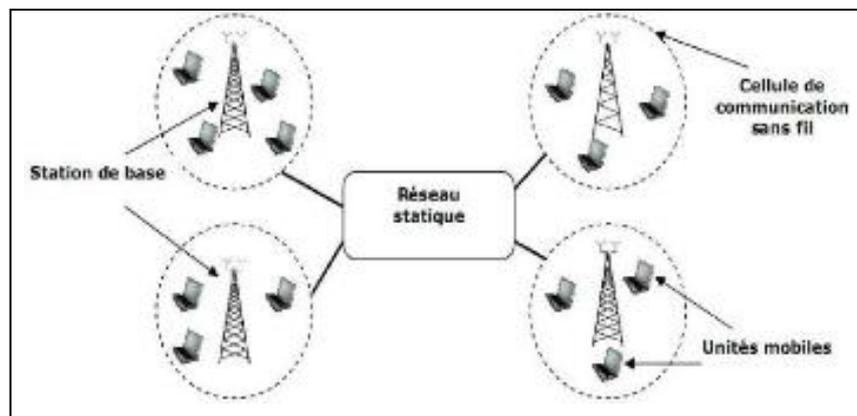
- **Réseaux cellulaires (avec infrastructure) :**

Ce type de réseaux se compose des éléments suivants :

Les "sites fixes" du réseau filaire.

Les "sites mobiles", réseaux sans fils.

Certains sites fixes, appelés stations de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites mobiles localisés dans une zone géographique limitée, appelée cellule comme le montre la figure suivante [2]



**Figure 2 des réseaux mobiles avec infrastructure [17]**

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire. Une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée.[17]

- **Réseaux ad hoc (sans infrastructure) :**

L'évolution récente de la technologie dans le domaine de la communication sans fil et l'apparition des unités de calculs portables (les laptops par exemple),

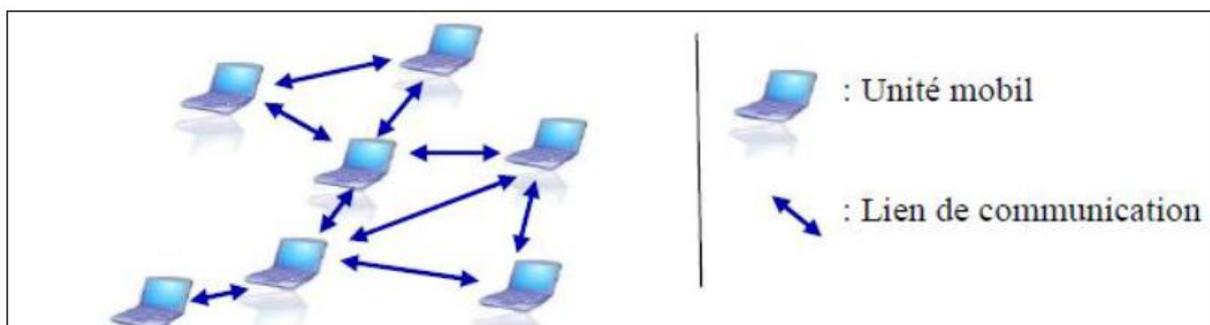
poussent aujourd'hui les chercheurs à faire des efforts afin de réaliser le but des réseaux :

*"L'accès à l'information n'importe où et n'importe quand"*

Le concept des réseaux mobiles ad hoc essaie d'étendre les notions de la mobilité à tous les composants de l'environnement. Ici, contrairement aux réseaux sans fil basés sur les points d'accès (avec infrastructure), aucune administration centralisée n'est disponible, ce sont les hôtes mobiles elles-mêmes qui forment, d'une manière ad hoc, une infrastructure du réseau. Aucune supposition ou limitation n'est faite sur la taille du réseau ad hoc, le réseau peut contenir des centaines ou des milliers d'unités mobiles. Les réseaux ad hoc sont idéals pour les applications caractérisées par une absence (ou la non fiabilité) d'une infrastructure préexistante, tel que les applications militaires et les autres applications de tactique comme les opérations de secours (incendies, tremblement de terre..) et les missions d'exploration [18]

- **Définition d'Un réseau ad hoc:**

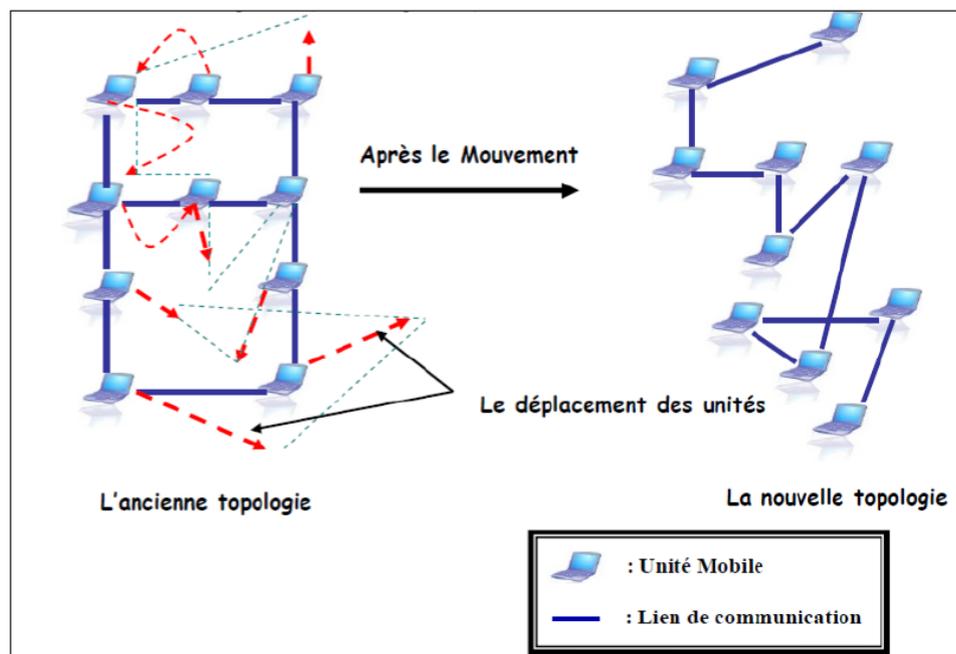
Un réseau ad hoc, connu aussi sous le nom de MANET (Mobile Ad hoc Network), est un réseau dont la topologie ne bénéficie d'aucune infrastructure fixe préexistante ou administration topologie ne bénéficie d'aucune infrastructure fixe préexistante ou administration centralisée [3].ca se représenté dans la figure 3 :



**Figure 3 Réseau en mode ad hoc [18]**

Ici, contrairement aux réseaux basés sur la communication cellulaire, ce sont les unités mobiles elles même qui forment, d'une manière ad hoc, une infrastructure du réseau et en maintiennent la connectivité d'une façon décentralisée. L'information est transmise par l'intermédiaire des mobiles présents. Un réseau ad hoc est un réseau sans fil auto-configurable. Lorsque deux machines mobiles ou plus se retrouvent dans le même secteur géographique, elles doivent se reconnaître pour pouvoir s'échanger des données. Le réseau doit se configurer automatiquement ; périodiquement ou à la demande pour assurer la liaison entre ces machines [3].

Un réseau ad hoc doit être facilement déployé, les nœuds peuvent joindre et quitter le réseau de façon totalement dynamique sans devoir en informer le réseau et si possible sans perturber les communications entre les autres nœuds du réseau. La topologie des réseaux ad hoc est dynamique et décentralisé. Elle peut changer aléatoirement, et les unités mobiles sont libres de se déplacer arbitrairement. De ce fait, la déconnexion de ces unités est très fréquente [17],ce ci est présenté dans la figure suivante



**Figure 4 Le changement de la topologie des réseaux ad hoc[18]**

- **Caractéristiques des réseaux ad hoc :**

- **Mobilité (Une topologie dynamique) :**

La mobilité des nœuds constitue à l'évidence une caractéristique très spécifique des Réseaux ad hoc. Cette mobilité est intrinsèque au fonctionnement du réseau. Dans un réseau ad hoc, la topologie du réseau peut changer rapidement, de façon aléatoire et non prédictible et les techniques de routage des réseaux classiques, basées sur des routes préétablies, ne peuvent Plus fonctionner correctement [18].

- **Equivalence des nœuds du réseau :**

Dans un réseau classique, il existe une distinction nette entre les nœuds terminaux (Stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données.

Cette différence n'existe pas dans les réseaux ad hoc car tous les nœuds peuvent être amenés à assurer des fonctions de routage [19].

- **Liaisons sans fil (Une bande passante limitée) :**

Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau ad hoc. Malgré des progrès très importants, leurs performances restent et resteront en dessous de celles des technologies des réseaux filaires. La bande passante est moins importante, alors que le routage et la gestion de la mobilité génèrent davantage de flux de contrôle et de signalisation que dans une architecture de réseau filaire. Ces flux doivent être traités de façon prioritaire pour prendre en compte rapidement les modifications de topologie [18].

- **Autonomie des nœuds (Des contraintes d'énergie) :**

La consommation d'énergie constitue un problème important pour des équipements fonctionnant grâce à une alimentation électrique autonome. Ces équipements intègrent des modes de gestion d'énergie et il est important que les protocoles mis en place dans les réseaux ad hoc prennent en compte ce problème [4].

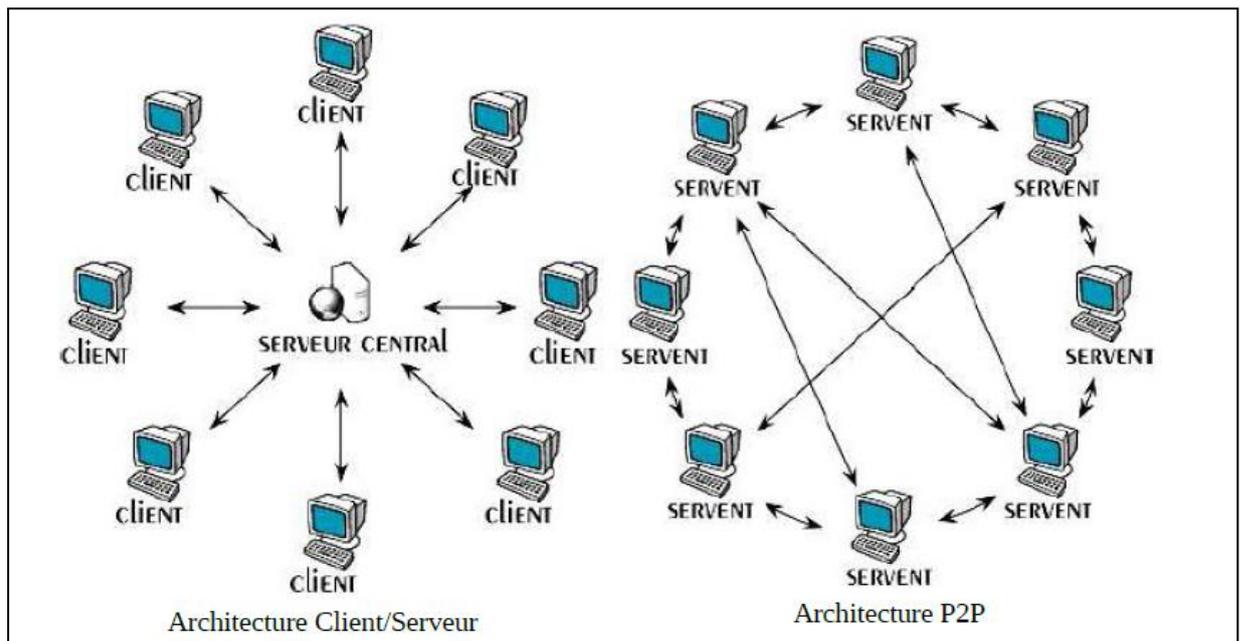
### ➤ **Vulnérabilité (Une sécurité physique limitée) :**

Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux ad hoc, le principal problème ne se situe pas au niveau du support physique mais principalement dans le fait que tous les nœuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau [4]

## **I.3 Architecture client/serveur :**

### **I.3.1 Principe du client/serveur**

- **Un client :** Les caractéristiques d'un client sont les suivantes : il est d'abord actif il envoie des requêtes au serveur, il attend et reçoit les réponses du serveur.
- **Un serveur :** Un serveur est initialement passif, il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients. Dès qu'une requête lui parvient, il la traite et envoie une réponse.
- **Le dialogue :** Le client et le serveur doivent bien sûr utiliser le même protocole de communication. Un serveur est généralement capable de servir plusieurs clients simultanément.[15]
- **Remarques :** Une fois le client traité, le serveur peut en traiter un autre. Il existe des serveurs multi clients comme les serveurs Web/http qui sont capables de traiter plusieurs clients en même temps. Il existe aussi des serveurs « non connectés », dans ce cas il n'y a pas de connexion ou de déconnexion. Un autre type d'architecture réseau est le pair à pair (peer to peer en anglais ou P2P), dans lequel chaque ordinateur ou logiciel est à la fois client et serveur. [19]



**Figure 5 architecture client/serveur et architecture p2p [19]**

Les systèmes pair-à-pair permettent à plusieurs ordinateurs de communiquer via un réseau, de partager simplement des fichiers le plus souvent, mais également des flux multimédia ou encore un service (comme la téléphonie avec Skype par exemple), ... sur internet. L'utilisation d'un système pair-à-pair nécessite pour chaque nœud l'utilisation d'un logiciel particulier. Ce logiciel, qui remplit alors à la fois les fonctions de client et de serveur, est parfois appelé *servent* (de la contraction de « serveur » et de « client »), ou plus communément mais de façon réductrice, « client ».

Une des applications la plus connue est le partage de fichiers par le biais de logiciel à la fois client et serveur comme e Donkey, eMule, FastTrack (utilisé par KaZaA) ou BitTorrent... Les serveurs pair-à-pair fonctionnent dans la quasi-totalité des cas en mode synchrone : le transfert d'information est limité aux éléments connectés en même temps au réseau. Ils peuvent utiliser le protocole TCP (Transmission Control Protocol ou protocole de contrôle de transmissions) comme couche de transport des données (il fonctionne en duplex, la réception des données est donc confirmée et leur intégrité est assurée).[14]

---

En revanche, certaines utilisations comme le continu (streaming) nécessitent l'emploi d'un protocole plus léger et plus rapide, comme UDP (User Datagram Protocol ou protocole de datagramme utilisateur), bien que moins fiable, l'UDP est aussi le protocole le plus utilisé pour transmettre des messages entre serveurs dans les systèmes en partie centralisés.[15]

### **I.3.2 Communication client/serveur :**

Le mode client/serveur n'est pas le modèle de communication parfait, il n'y en a pas ! Connaissant les avantages et les inconvénients par rapport au mode distribué (par exemple pair à pair), il est possible de choisir celui qui convient le mieux.[14]

### **I.3.3 Avantages :**

Toutes les données sont centralisées sur un seul serveur, on a donc « un contrôle de sécurité simplifié ». Les technologies supportant l'architecture client/serveur sont plus matures que les autres (et plus anciennes). L'administration se porte au niveau serveur. Toute la complexité/puissance peut être déportée sur le(s) serveur(s), les utilisateurs utilisant simplement un client léger.

Les serveurs étant centralisés, cette architecture est particulièrement adaptée et vélocité pour retrouver et comparer de vastes quantités d'information (moteur de recherche sur le web).[20]

### **I.3.4 Inconvénients**

Si trop de clients veulent communiquer sur le serveur en même temps, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair à pair fonctionnent mieux en ajoutant de nouveaux participants).

Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair à pair continue à fonctionner, même si plusieurs participants quittent le réseau). Les coûts de mise en place et de maintenance sont élevés. En aucun cas les clients ne peuvent communiquer entre eux, entraînant une asymétrie de

L'information au profit des serveurs.[15]

## I.4 Java Socket Programmation

La programmation Java Socket est utilisée pour la communication entre les applications s'exécutant sur différents JRE.

Les classes Socket et ServerSocket sont utilisées pour la programmation de socket.

Le client en programmation socket doit connaître deux informations :

- Adresse IP du serveur,
- et Numéro de port.

Cette figure définit l'architecture client serveur [20]

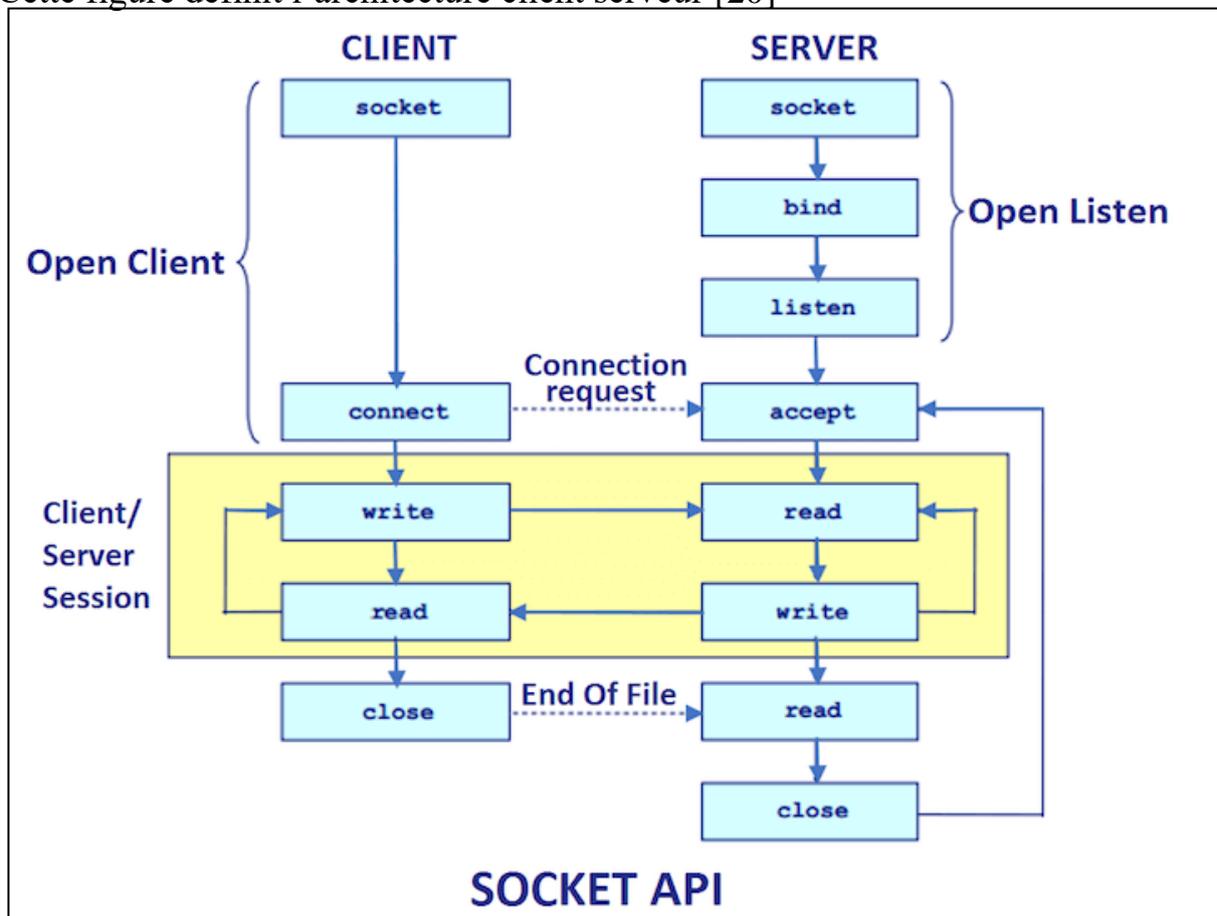


Figure 6 Socket API [20]

## I.4.1 classe de socket

Un socket est simplement un point de terminaison pour les communications entre les machines. La classe Socket peut être utilisée pour créer un socket.[15]

### I.4.1.1 Méthodes importantes

Méthode	Description
1) <code>public InputStream getInputStream()</code>	renvoie le <code>InputStream</code> attaché à ce socket.
2 <code>publics OutputStream getOutputStream</code>	renvoie le <code>OutputStream</code> attaché à ce socket.
3) <code>public synchronized void close()</code>	fermer cette prise

### I.4.1.2 Classe ServerSocket

La classe ServerSocket peut être utilisée pour créer un socket serveur. Cet objet est utilisé pour établir la communication avec les clients ;[15]

### I.4.1.3 Méthodes importantes :

Méthode	Description
1) <code>public Socket accept()</code>	renvoie le socket et établit une connexion entre le serveur et le client.
2) <code>public synchronized void close()</code>	ferme le socket du serveur.

## I.4.2 Exemple de programmation de socket

### Java :

- **Création du srveur :**

Pour créer l'application serveur, nous devons créer l'instance de la classe ServerSocket. Ici, nous utilisons le numéro de port 6666 pour la communication

entre le client et le serveur. Vous pouvez également choisir n'importe quel autre numéro de port. La méthode accept() attend le client. Si les clients se connectent avec le numéro de port donné, il renvoie une instance de Socket.[15]

```
ServerSocket ss=new ServerSocket(6666);  
Socket s=ss.accept();//établit la connexion et attend le client
```

- **Création du client :**

Pour créer l'application cliente, nous devons créer l'instance de la classe Socket. Ici, nous devons transmettre l'adresse IP ou le nom d'hôte du serveur et un numéro de port. Ici, nous utilisons "localhost" car notre serveur fonctionne sur le même système.

```
Socket s=new Socket("localhost",6666);
```

Voyons un exemple simple de programmation de socket Java où le client envoie un texte et le serveur le reçoit et l'imprime.

- File: MyServer.java

```
import java.io.*;  
import java.net.*;  
public class MyServer {  
public static void main(String[] args){  
try{  
ServerSocket ss=new ServerSocket(6666);  
Socket s=ss.accept();//establishes connection  
DataInputStream dis=new DataInputStream(s.getInputStream());  
String str=(String)dis.readUTF();  
System.out.println("message= "+str);  
ss.close();  
}catch(Exception e){System.out.println(e);}
```

```
}}
```

- File: MyClient.java

```
import java.io.*;
import java.net.*;
public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
dout.writeUTF("Hello Server");
dout.flush();
dout.close();
s.close();
}catch(Exception e){System.out.println(e); } } }
```

## I.5 Conclusion :

L'essor des technologies sans fil, offre aujourd'hui de nouvelles perspectives dans le domaine des télécommunications. L'évolution récente des moyens de la communication sans fil a permis la manipulation de l'information à travers des unités de calculs portables qui ont des caractéristiques particulières (une faible capacité de stockage, une source d'énergie autonome..) et accèdent au réseau à travers une interface de communication sans fil.

---

## **II. Chapitre 2 : Android**

## II.1 Introduction :

Android est un système d'exploitation open source pour terminaux mobiles créé à la base par une startup du même nom et qui a été rachetée par la firme Google en août 2005 : Cette appellation tient ses origines du terme « Androïde » qui désigne un robot sous forme humaine.

En novembre 2007, Google créa l'OHA (Open Hand Alliance), un groupe de plus d'une cinquantaine d'entreprises de divers domaines (opérateurs mobiles, fabricants de téléphones et de semi-conducteurs, éditeurs de logiciels, distributeurs,...) à l'instar de HTC, Sony, DELL, Intel, Samsung, LG,... Son objectif étant de promouvoir et de développer des standards open sources pour appareils mobiles, ce qui lui a permis d'aboutir à la réalisation d'Androïde.

Android est passé par plusieurs versions tout au long de son processus de développement. Depuis la version 1.5, Google a adopté la convention d'attribuer un nom de code à chacune de ses versions en suivant un ordre alphabétique et en faisant référence à un dessert (gâteaux, bonbons,...).

Il existe néanmoins quelques noms de codes qui sont associés à plusieurs versions successives tandis qu'il y en a d'autres liés à une seule. Les versions 1.0 et 1.1 n'ont pas été officiellement nommées bien que cette dernière admette en interne un nom de code «petit four».

Les principales versions d'Android ainsi que leurs dates d'apparition sont données par la liste suivante :

Android 1.0, API 1, septembre 2008; Android 1.1, API 2, février 2009; Android 1.5, API 3, avril 2009; Android 1.6, API 4, septembre 2009; Android 2.0-2.1, API 5-7, octobre 2009; Android, 2.2-2.2.3, API 8, mai 2010; Android 2.3-2.3.7, API 9-10, décembre 2010; Android, 3.0-3.2.6, API 11-13, février 2011; Android 4.0-4.0.4, API 14-15, octobre 2011; Android 4.1-4.3.1, API 16-18, juillet 2012; Android 4.4-4.4.4, API 19-20, octobre 2013; Android 5.0-5.1.1, API 21-22, novembre 2014; Android 6.0-6.0.1, API 23, octobre 2015; Android 7.0-7.1.2, API 24-25, août 2016; Android 8.0-8.1, API 26-27, août 2017; Android 9, API 28, juin 2018; Android 10, septembre 2019; Android 11, septembre 2020;[21]

## II.2 Logo

Le personnage nommé *Bugdroid* est le petit robot vert utilisé par Google pour présenter Android. Ce personnage est sous la licence « *Creative Commons by (3.0)* » et peut donc être utilisé librement.[8]

Le site *Engadget* annonce que Bugdroid, le logo d'Android, serait en fait un personnage d'un jeu des années 1990 sur Atari : *Gauntlet: The Third Encounter*.

En 2019 Google fait évoluer l'identité visuelle d'Android, et abandonne la numérotation des versions avec des noms de desserts pour plus de clarté et de diversité.[21]



**Figure 7 logo android 2007 /2014 [21]**



**Figure 8 logo android 2014/2019 [21]**



**Figure 9 Logo android 2019 [21]**

## II.3 Architecture d'Android

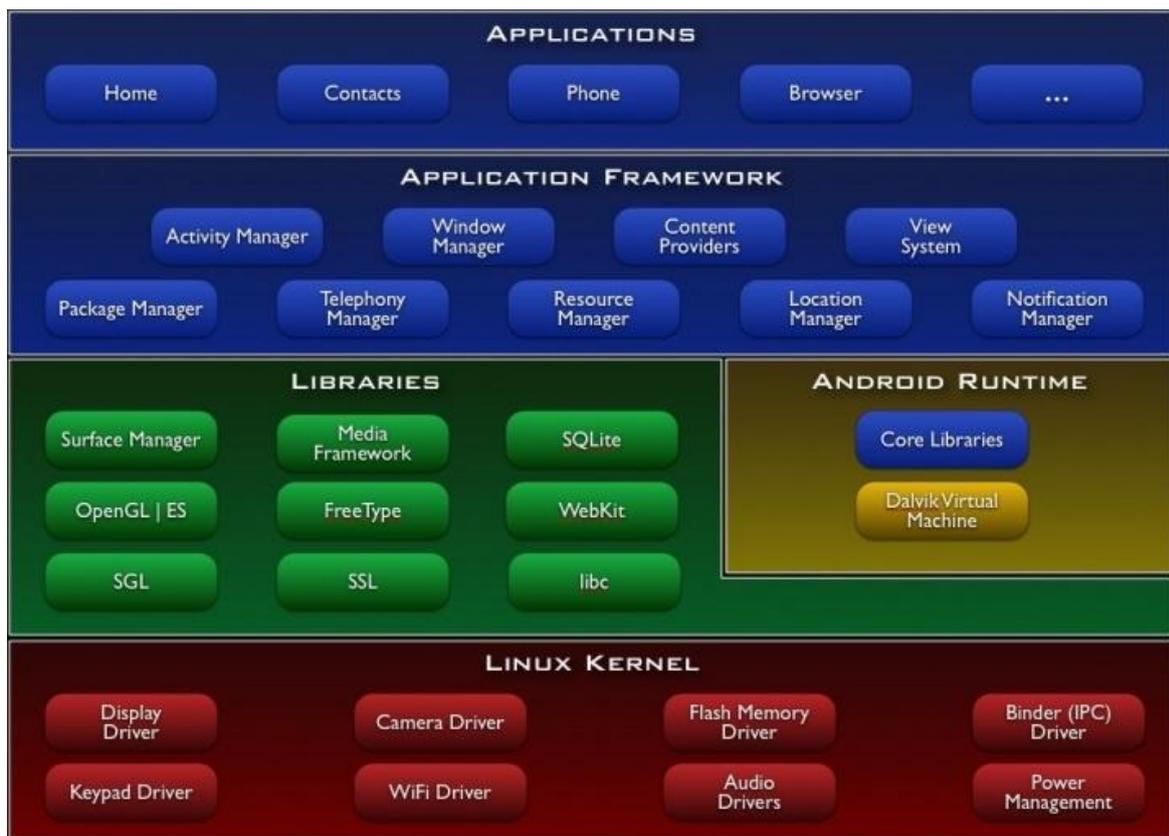


Figure 10 Architecture d'Android [22]

**Noyau linux** (*Linux Kernel*) : Android est basé sur la version 2.6 de Linux. Ce noyau permet la gestion de la mémoire, la gestion des processus et possède un modèle de sécurité basé sur un système de permissions utilisé depuis 1970.

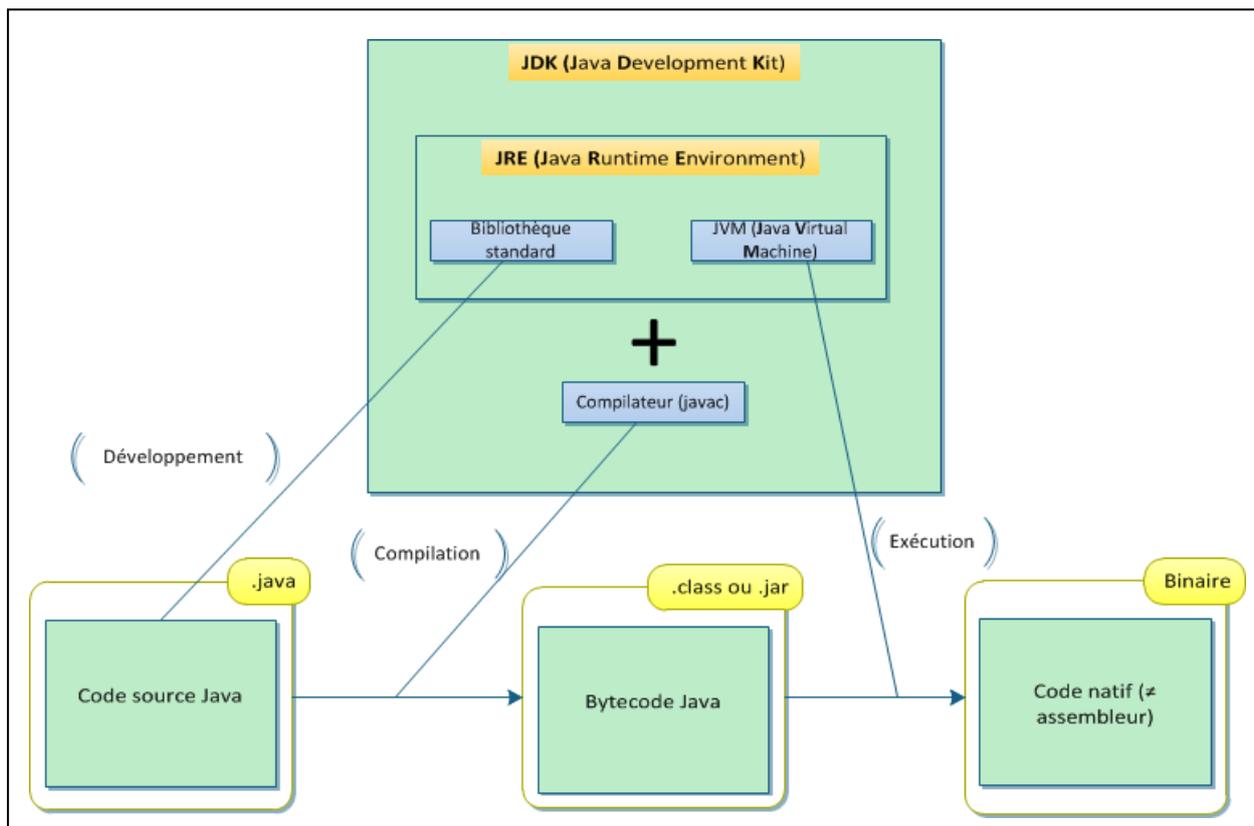
**Librairies** (*libraries*) : C'est un ensemble de bibliothèques natives écrites en C/C++ fournissant toutes les fonctionnalités de bas niveau. SQLite et WebKit sont deux exemples de ces bibliothèques[22]

**Moteur d'exécution Android** (*Android Runtime*) : il est composé d'un ensemble de bibliothèques de base du langage JAVA et de la machine virtuelle Dalvik ou ART.[22]

**Framework applicatif** (*Application Framework*) : il offre les classes nécessaires pour développer des applications Android. Cette couche contient un ensemble de gestionnaires (Manager) permettant l'accès aux matériels (capteurs, GPS, caméra,...) et aux services de base (fenêtrage, ressources, téléphonie, fournisseurs de contenu, notification,...).[22]

**Applications :** ce sont les applications installées par défaut telles que Calendrier, Calculatrice, Contact et les applications ajoutées par l'utilisateur.[23]

Le moteur d'exécution d'Android C'est cette couche qui fait qu'Android n'est pas qu'une simple « implémentation de Linux pour portables ». Elle contient certaines bibliothèques de base du Java accompagnées de bibliothèques spécifiques à Android et la machine virtuelle « Dalvik ». La figure suivante est un schéma qui indique les étapes nécessaires à la compilation et à l'exécution d'un programme Java standard.[7]



**Figure 11 compilation et à l'exécution d'un programme Java standard [23]**

Votre code est une suite d'instructions que l'on trouve dans un fichier **.java** qui sera traduit en une autre suite d'instructions dans un autre langage que l'on appelle le «bytecode». Ce code est contenu dans un fichier **.class**. Le bytecode est un langage spécial qu'une machine virtuelle Java peut comprendre et interpréter. Les différents fichiers **.class** sont ensuite regroupés dans un **.jar**, et c'est ce fichier qui est exécutable. En ce qui concerne Android, la procédure est différente. En fait, ce que vous appelez Java est certainement une variante particulière de Java qui s'appelle «Java SE». Or, pour développer des applications pour Android, on n'utilise pas vraiment Java SE. Pour ceux qui

savent ce qu'est «Java ME», ce n'est pas non plus ce framework que l'on utilise (Java ME est une version spéciale de Java destinée au développement mobile, mais pas pour Android donc).[23]

À noter que sur le schéma le JDK et le JRE sont réunis, mais il est possible de télécharger le JRE sans télécharger le JDK.[7]

## II.4 L'environnement Dalvik

**Dalvik** est une machine virtuelle destinée aux téléphones mobiles et tablettes tactiles, qui est incorporée dans le système d'exploitation Android. Dalvik est destiné à permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité (peu d'espace mémoire et peu de puissance de calcul).

Créé par Dan Bornstein, Dalvik fait partie du runtime, le moteur qui permet l'exécution des applications pour Android. Le nom est celui d'un village de pêcheurs en Islande, pays d'origine de son auteur.

Dalvik offre une alternative à la machine virtuelle de la technologie Java. Les fichiers de bytecode de la technologie Java sont préalablement transformés et consolidés dans un fichier `.dex` (*Dalvik Executable*) en vue de leur utilisation par Dalvik.[23]

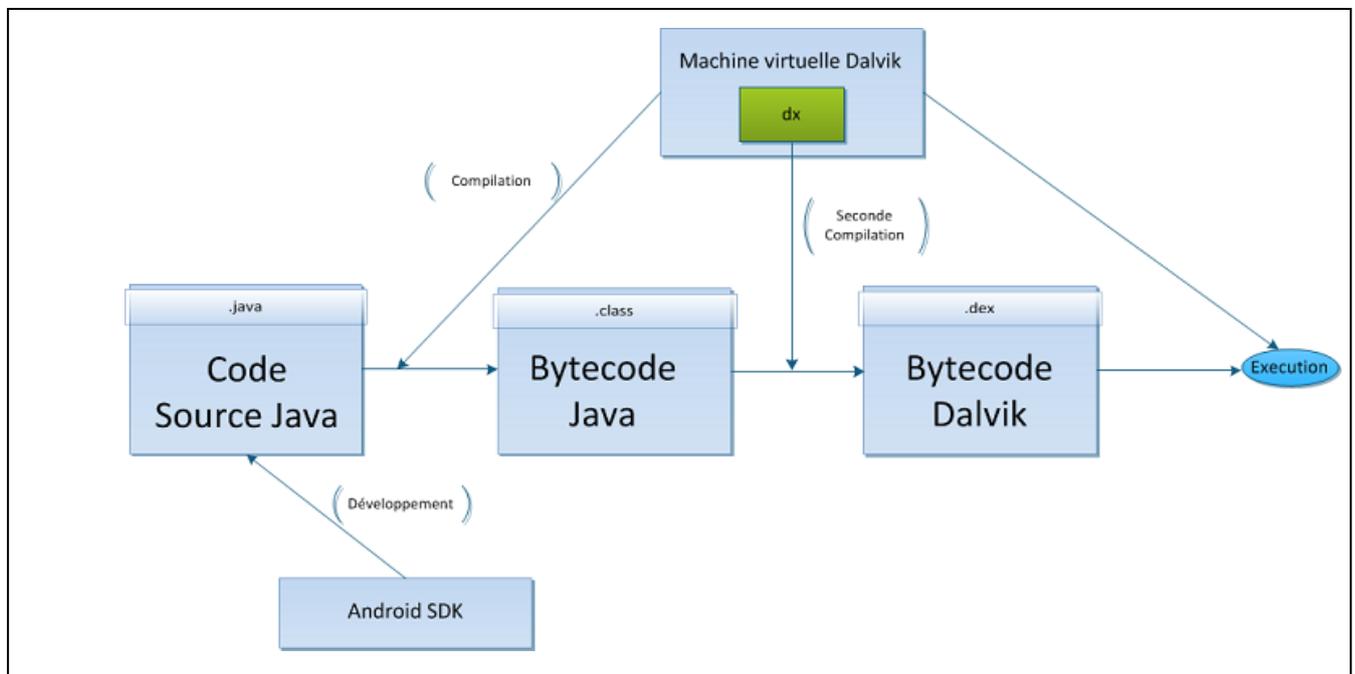


Figure 12 la compilation et à l'exécution d'un programme Android standard[23]

---

## II.5 Cycle de vie d'une activité (Activity Lifecycle)

Pour développer d'une application sur Android, on doit comprendre le cycle de vie d'une activité.[28]

- **L'état Active/courant (Running)** : l'activité se trouve au premier plan, et reçoit les interactions utilisateurs. Si l'appareil a besoin des ressources, l'activité se trouvant en bas de la back est tuée.
- **L'état Paused (il est en pause)** : L'activité est visible mais l'utilisateur ne peut pas interagir avec. La seule différence avec l'état précédent est la non-réception des événements utilisateurs.
- **L'état Stopped** : L'activité n'est pas visible mais toujours en cours d'exécution. Toutes les informations relatives à son exécution sont conservées en mémoire.
- **L'état Dead** : l'activité est tuée, elle n'est plus en cours d'exécution, et disparaît de la back stack.
- **Back stack** : toutes les activités sont stockées dans une liste que l'on appelle généralement back stack. Il existe trois boucles principales: Lors de la création d'activité, la méthode onCreate est appelée. Cet appel est suivi par la méthode onStart afin de signifier le lancement effective de l'application. Puis la méthode onResume est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité. Ces traitement devront être arrêtés lors d'appel à la méthode onPause est relancés si besoin lors d'un futur appel à la méthode onResume[8]

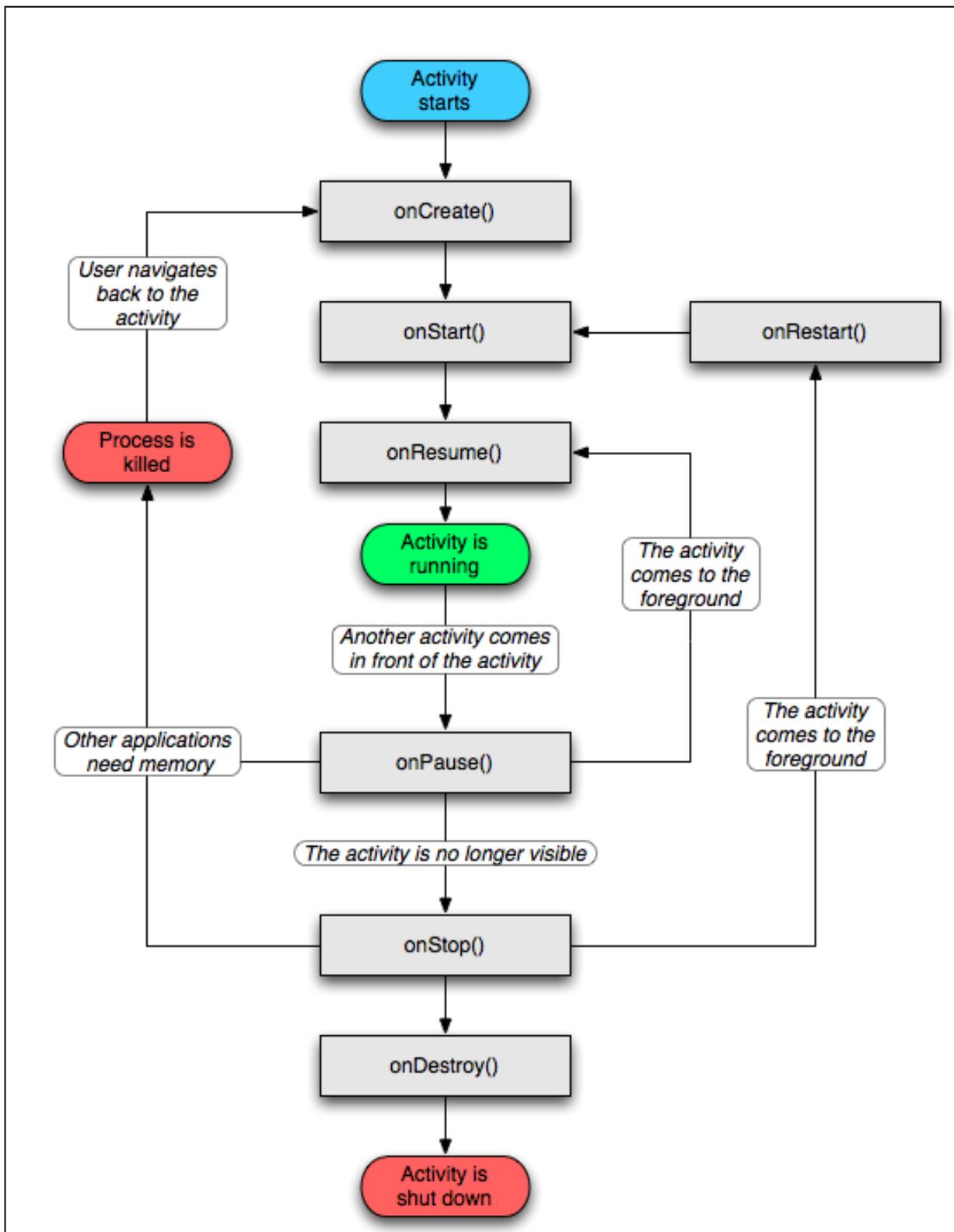


Figure 13 cycle de vie d'une activité android [24]

---

## **II.6 Conclusion:**

En générale dans ce chapitre on a définir le système android ;

Android est le système d'exploitation mobile crée par Google. Il équipe la majorité des téléphones portables du moment (smartphones). Son principal concurrent est Apple avec l'iPhone. Android est un système vous permettant de personnaliser votre téléphone, télécharger des applications (navigateur Internet, GPS, Facebook...). Android équipe également les tablettes tactiles.

---

### **III. Chapitre3 : La conception**

## III.1 Introduction :

Dans la première partie de ce chapitre, nous allons faire la conception de notre application gestion de rendez vous en utilisant le langage UML.

La deuxième partie de chapitre consacré à la présentation de la mise en œuvre de notre application.

## III.2 Présentation d'UML :

La programmation par objet prend de l'importance, il devient nécessaire de trouver une méthode qui lui soit adaptée. De cet effet plusieurs méthodes sont apparues telles que:

- OMT (Object Modeling Technique) de James Rumbaugh (General Electric) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système.
- OOD (Object Oriented Department) de GradyBooch, définie pour le Département of Defense, introduit le concept de paquetage (package).
- OOSE (Object Oriented Software Engineering) d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

Les constructeurs de ces trois méthodes se mirent d'accord pour définir une méthode commune, par l'unification de leurs méthodes. UML est né de cet effort de convergence. En fait, et comme son nom l'indique, UML (UnifiedModelingLanguage) est un langage de modélisation objet et non pas une méthode objet. En conséquence, la notation UML a été conçue pour servir de langage de modélisation objet, indépendamment de la méthode mise en oeuvre.

L'unification a progressé par étapes. En 1995, Booch et Rumbaugh se sont mis d'accord pour construire une méthode unifiée, UnifiedMethod 0.8, en 1996, Jacobson les a rejoints pour produire UML 0.9 (remplacement du mot méthode par le mot langage). Les acteurs les plus importants dans le monde du logiciel s'associent alors à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational , Unisys etc.) Et UML 1.0 est soumis à l'OMG (Object Managment Group). Elle adopte en novembre 1997 UML 1.1 comme langage de modélisation des systèmes d'information à objets. La version d'UML en cours en 2008 est UML 2.1.1 et aujourd'hui les travaux d'amélioration se poursuivent [9].

## III.2.1 Types de diagrammes UML

L'UML utilise des éléments et les associe de différentes manières pour former des diagrammes qui représentent les aspects statiques ou structurels d'un système, ainsi que des diagrammes comportementaux qui capturent les aspects dynamiques d'un système.

### III.2.1.1 Diagrammes UML structurels

- **Diagramme de classes** : Diagramme UML le plus couramment utilisé et fondement de toute solution orientée objet. Classes d'un système, attributs et opérations, et relations entre chaque classe. Les classes sont regroupées pour créer des diagrammes de classe lors de la modélisation de systèmes de grande taille.
- **Diagramme de composants** : Représente la relation structurelle entre les éléments d'un système logiciel, le plus souvent utilisé avec des systèmes complexes disposant de multiples composants. Les composants communiquent à l'aide d'interfaces.
- **Diagramme de structure composite** : Les diagrammes de structure composite sont utilisés pour présenter la structure interne d'une classe.
- **Diagramme de déploiement** Illustre les infrastructures physique et logicielle d'un système. Ils sont utiles lorsqu'une solution logicielle est déployée sur de nombreuses machines avec des configurations uniques.
- **Diagramme d'objets** Montre les relations entre des objets à travers des exemples tirés du monde réel et permet de voir l'apparence d'un système à n'importe quel instant donné. Les données sont disponibles à l'intérieur des objets, elles peuvent donc être utilisées pour clarifier les relations entre des objets.
- **Diagramme de paquetages** Il existe deux types de dépendances spéciales entre les paquetages : l'importation de paquetages et la fusion de paquetages. Les paquetages peuvent représenter les différents niveaux d'un système pour en révéler l'architecture. Les dépendances des paquetages peuvent être marquées pour mettre en avant le mécanisme de communication entre les différents niveaux.

### ***III.2.1.2 Diagrammes UML comportementaux***

- **Diagrammes d'activités** Flux de travail d'entreprise ou opérationnels représentés graphiquement pour montrer l'activité de chacune des composantes du système. Les diagrammes d'activités sont utilisés comme une alternative aux diagrammes états-transitions.
- **Diagramme de communication** Semblable à un diagramme de séquence, mais l'accent est mis sur les messages transmis entre les objets. La même information peut être représentée par un diagramme de séquence et différents objets.
- **Diagramme global d'interaction** Il existe sept types de diagrammes d'interaction, et ce diagramme montre l'ordre dans lequel ils agissent.
- **Diagramme de séquence** Montre comment les objets interagissent les uns avec les autres et dans quel ordre. Ils représentent les interactions d'un scénario particulier.
- **Diagramme états-transitions** Semblable à un diagramme d'activités, il décrit le comportement des objets qui se comportent de diverses manières dans leur état actuel.
- **Diagramme de temps** À l'instar d'un diagramme de séquence, le comportement des objets est représenté sur un laps de temps donné. S'il y a un seul objet, le schéma est simple. S'il y a plus d'un objet, les interactions des objets sont présentées sur ce laps de temps précis.
- **Diagramme de cas d'utilisation** Il représente une fonctionnalité spécifique dans un système et est créé pour illustrer comment différentes fonctionnalités sont interconnectées et montrer leurs contrôleurs (ou acteurs) internes et externes. [10]

### **III.3 La conception :**

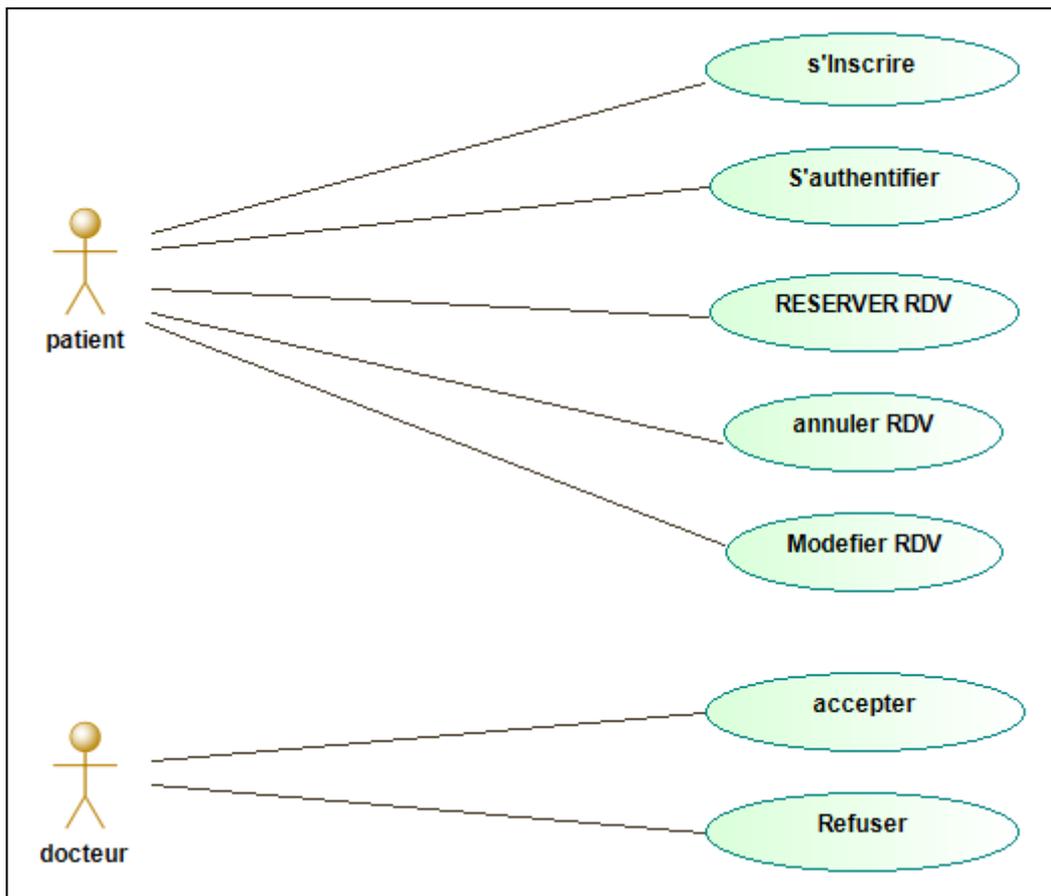
Dans cette partie, nous avons déterminé un premier découpage de l'application, et établit les différentes étapes de conception comme nous l'avons déjà cité. Les diagrammes appropriés pour notre projet sont :

- Diagramme de cas d'utilisation qui nous aide dans la spécification des besoins de notre système.

- Diagramme de séquence qui nous donne le scénario des actions et interactions entre les acteurs et notre système.
- Diagramme de classes qui définit l'architecture statique de notre système.

### III.3.1 Diagramme de cas d'utilisation :

La figure suivante représente le diagramme de cas d'utilisation pour le système Gestion de rendez-vous :



**Figure 14 Digramme de cas d'utilisation**

#### ➤ Identification Les acteurs :

Un acteur est une personne interagissant avec le système. Chaque acteur est identifié par un rôle. Ces rôles décrivent les capacités et les besoins de l'acteur. Les personnes interagissant avec le système Gestion de rendez vous sont :

- Le Patient.
- Le Docteur.

### ➤ Description des interactions :

- **S'inscrire** : le client doit remplir un formulaire. Ce formulaire nous servira à l'enregistrer sur notre base de données et de le contacter en cas de besoin (modification, annulation, validation, ...).
- **S'authentifier** : L'authentification se fait qu'une seule fois. Le client aura un identifiant et un mot de passe. La prochaine connexion sur le site sera prise en compte avec son identifiant.
- **Modifier** : Le client a le droit également de faire des modifications : Il peut modifier l'heure, la date, le lieu de son rendez-vous, en parallèle le système vérifiera si les modifications sont possibles (date, heure, lieu libres) c'est juste après qu'il aura le choix de faire ou non sa modification.
- **Annuler** : quand le client modifier le rendez vous le RDV sa sera annuler automatiquement
- **Accepter /Refuser** :  
Le statut du rendez-vous est mis par défaut « En attente ». Cette attente peut durer quelques jours avant que l'administratif système le valide. Le client recevra alors un message de confirmation du rendez-vous.

### III.3.2 Diagramme de séquence :

Dans notre projet, on a définis trois diagrammes de séquence. Ces diagrammes représentent toutes les actions et interactions entre acteurs et système selon un ordre chronologique

### III.3.2.1 L'inscription :

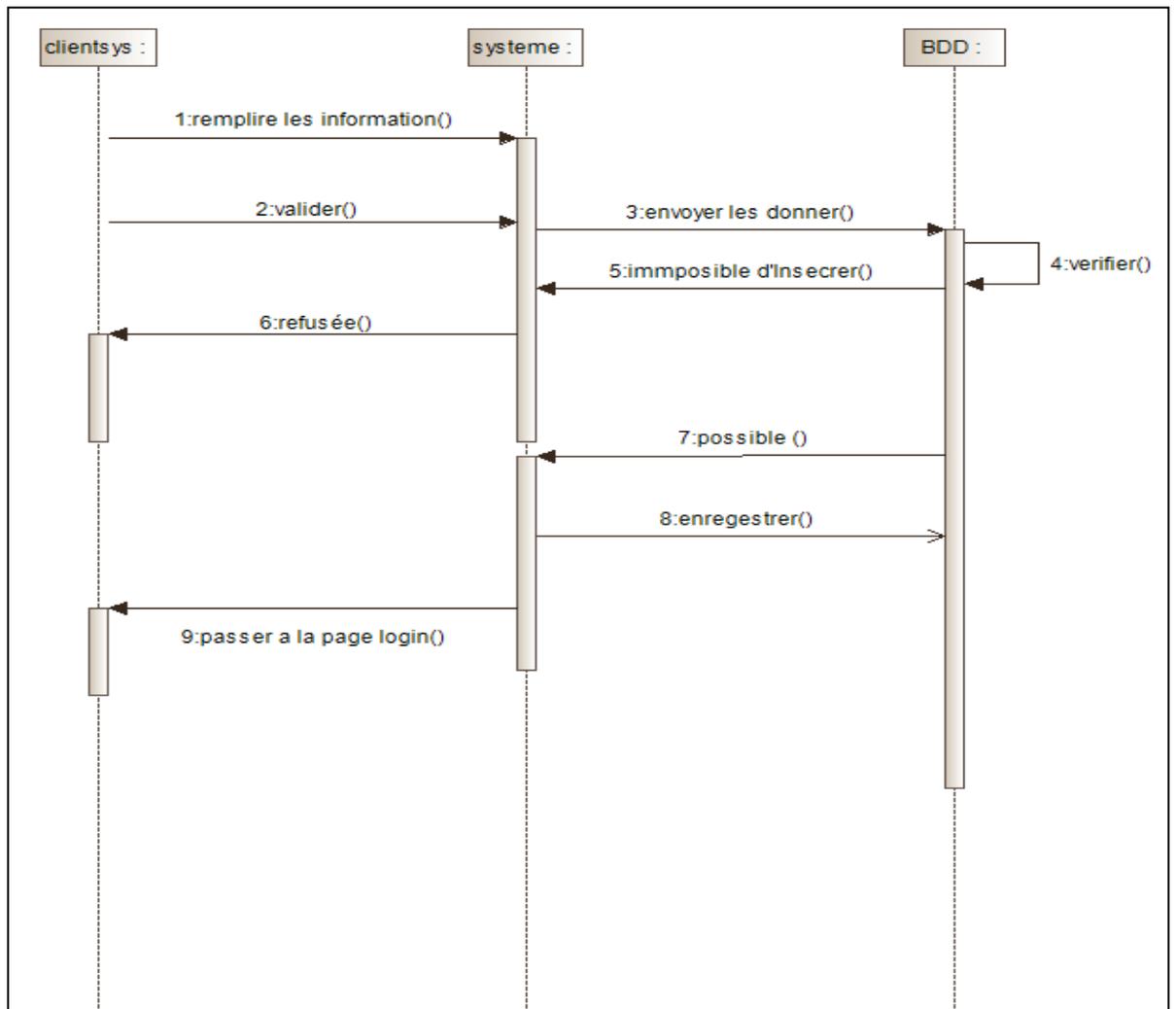


Figure 15 diagramme de séquence de l'inscription

### III.3.2.2 Connexion :

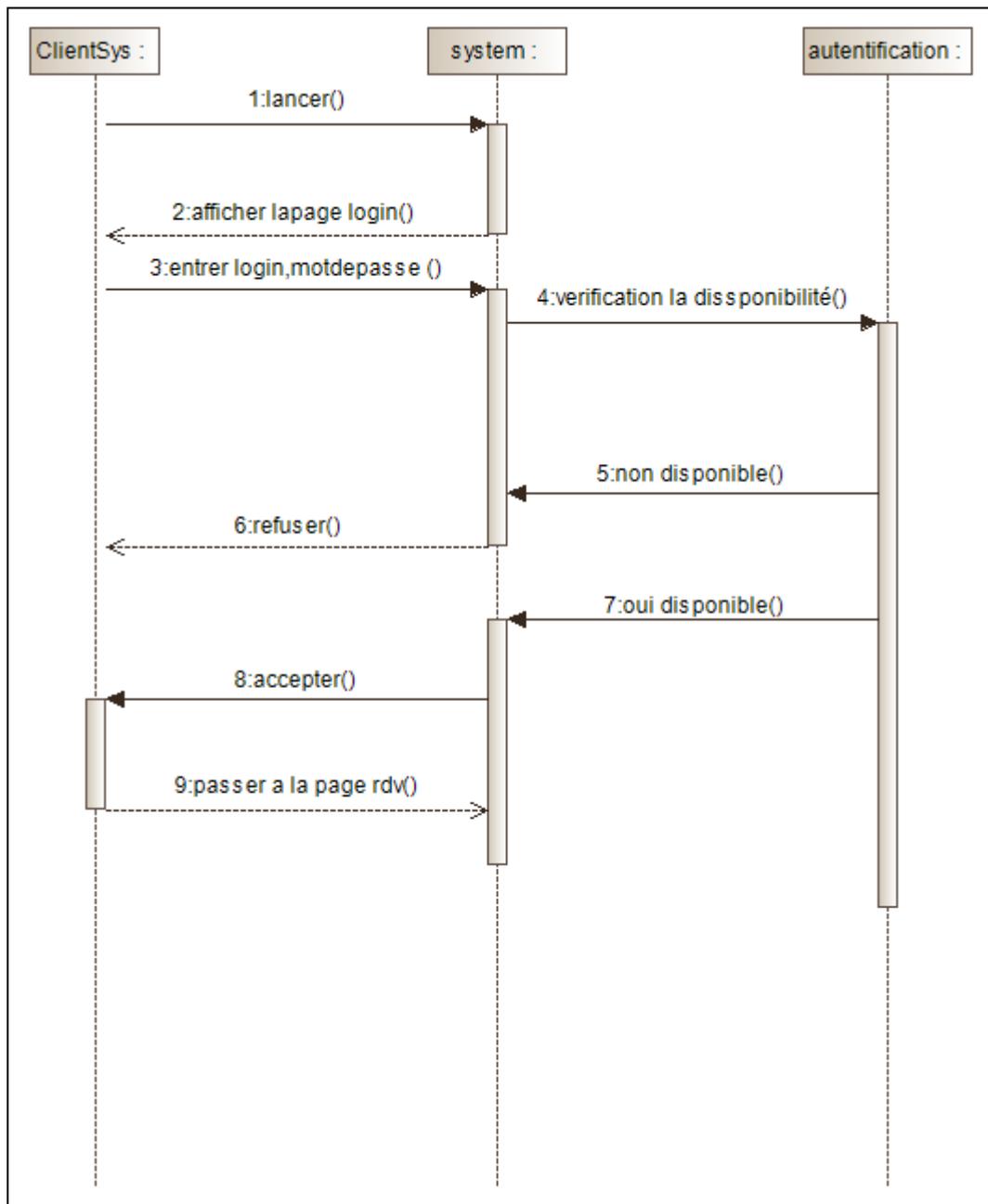


Figure 16 diagramme de séquence de Connexion

### III.3.2.3 Réserveation :

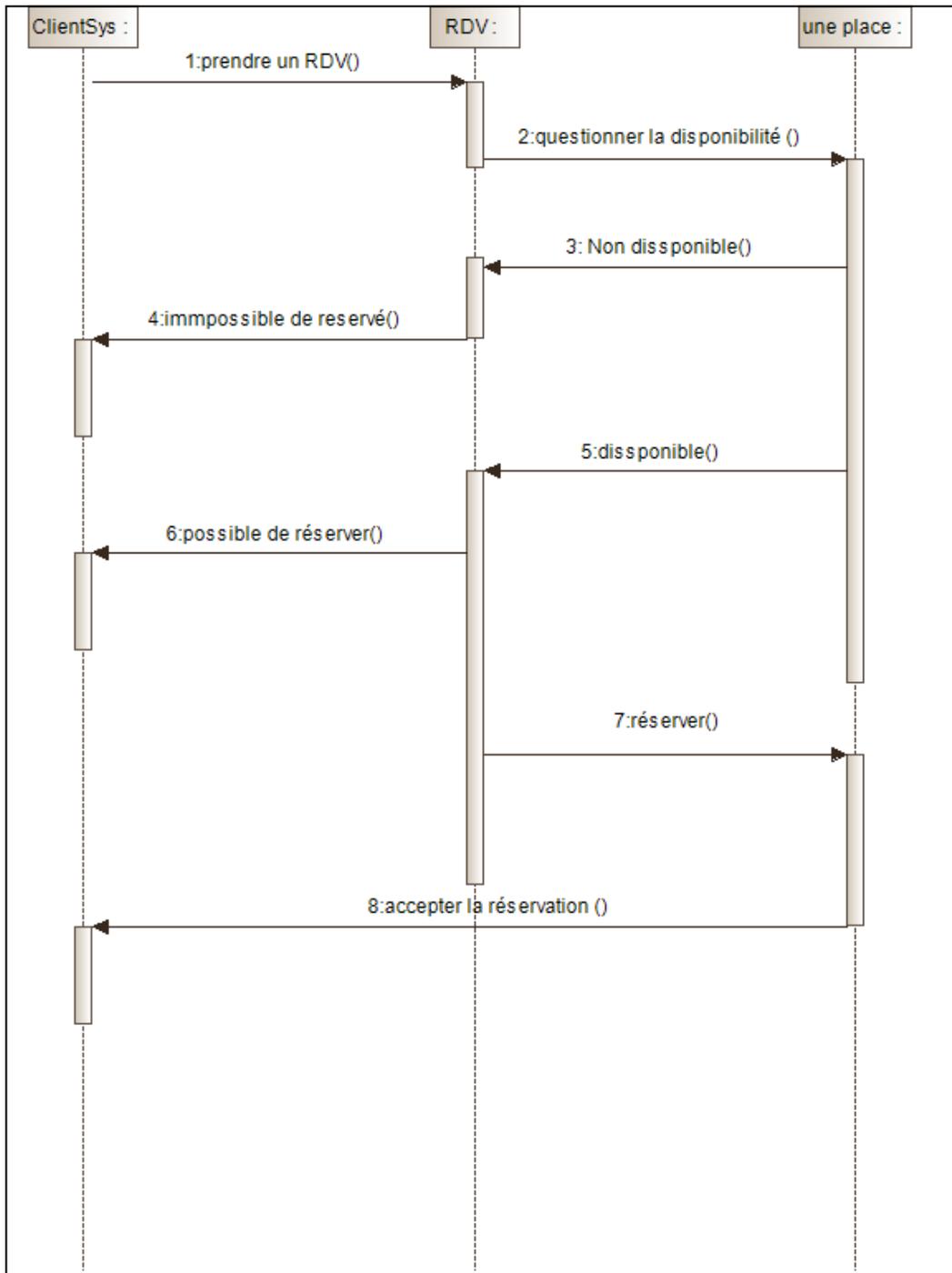


Figure 17 diagramme de séquence de Réserveation

### III.3.2.4 Diagramme de classe :

Le diagramme de classes fait abstraction des aspects temporels ou dynamiques. Il identifie les classes et les associations de notre système.

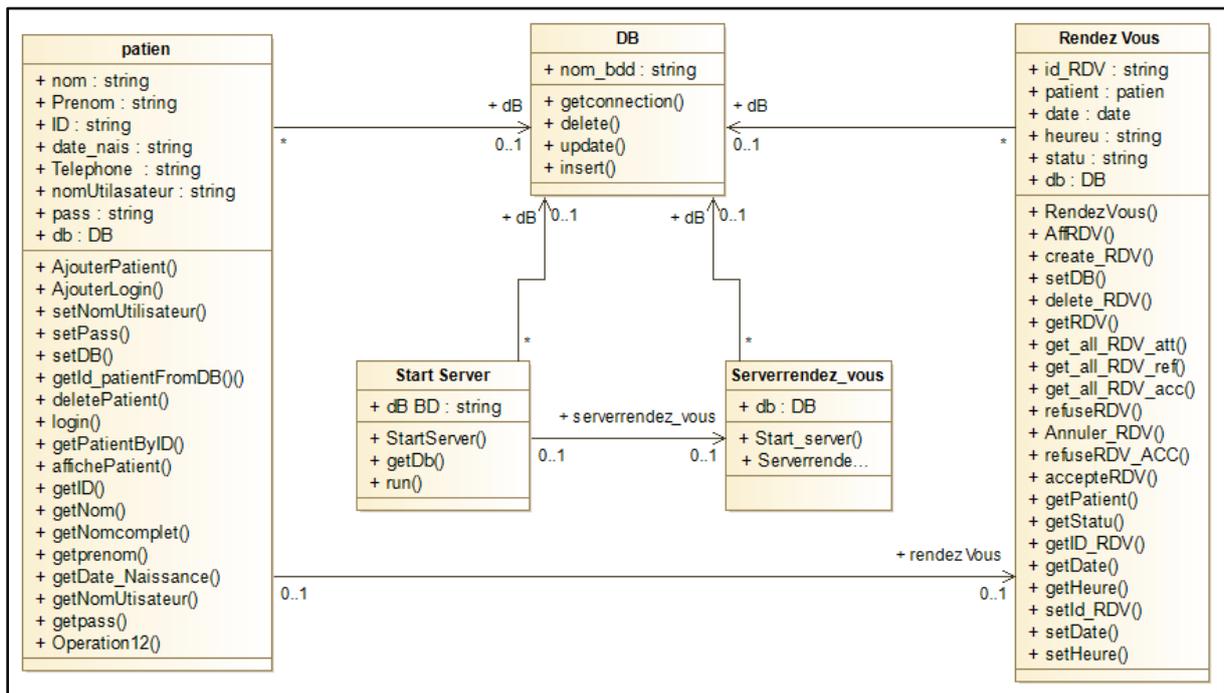


Figure 18 diagramme de Classe

### III.4 Conclusion :

Dans ce chapitre on a présentée le diagramme UML de notre project : diagramme de classe, de séquence et de cas d'utilisation pour bien définit le fonctionnement de notre système.

---

## **IV. Chapitre 4 : Implémentation**

## **IV.1 Introduction:**

Après avoir achevé l'étape conception de l'application, on va entamer a la partie Réalisation et implémentation dans laquelle on s'assure que le système est prêt pour être exploité par les utilisateurs finaux.

## **IV.2 Environnements de développement :**

### **IV.2.1 Netbeans :**

est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Développent and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de greffons.[11]

### **IV.2.2 AndroidStudio :**

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux. Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions.

variées simultanément<sup>18,19</sup>. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur. [12]

### **IV.2.3 MYSQL :**

est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde<sup>3</sup>, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server. Son nom vient du prénom de la fille du cocréateur Michael Widenius, My (sv) (prononcer [my]). SQL fait référence au Structured Query Language, le langage de requête utilisé. MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains<sup>4</sup>. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010<sup>5,6</sup>. Depuis mai 2009, son créateur Michael Widenius a créé MariaDB (Maria est le prénom de sa deuxième fille) pour continuer son développement en tant que projet Open Source. [13]

### **IV.3 Langage de programmation : Java**

Nous avons choisi pour développer notre application le langage Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.[14].

### **IV.4 Description de l'application :**

Notre application c'est une application client serveur, le client est installé sur le Smartphone, du patient, et connecter au serveur a un réseau.

## IV.4.1 Coté serveur :

Voici le coté serveur :



**Figure 19 coté serveur**

Quand on a appui sur le bouton START Server le serveur nous donner une adresse IP cette adresse IP nous le donner a le client pour bien connecte avec notre serveur :



**Figure 20 Start Server**

Puis quand le client insérer dans notre application il va afficher dans la table patient avec un ID le nom le prenom et ca date de naissance et le numero de telephone :



The screenshot shows a web application interface with a title bar 'Gestion des rendez-vous medical' in red text. Below the title bar is a navigation menu with tabs: 'Server', 'Patients', 'Rendes-vous en attente', 'Rendes-vous accepté', and 'Rendes-vous refusé'. The 'Patients' tab is selected. Below the menu is a table with the following columns: 'ID', 'NOM', 'PRENOM', 'DATE\_NAISSANCE', 'TELEPHONE', and 'TELEPHONE'. The table is currently empty, displaying the message 'Aucun contenu dans la table' in the center.

Figure 21 la table patient

#### IV.4.2 Coté Client :

La première page c la page de connecter avec l adresse IP donner par le serveur :

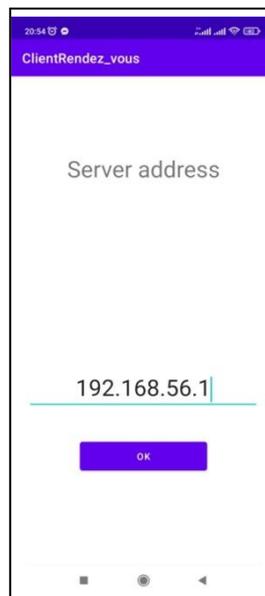


Figure 22 connecter avec serveur

Ensuit, le client passé à la page de login (connexion) :

Si le client déjà inscrit il peut entrer son nom d'utilisateur et son mot et passer la page suivante pour prendre un rendez vous



The screenshot shows a mobile application interface for a login screen. At the top, there is a purple header bar with the text "ClientRendez\_vous". Below the header, the word "Connexion" is displayed in a large, grey font. Underneath, there are two input fields: "Nom d'utilisateur" with the text "medjbar.98" and "Mot de passe" with the text "112233". A blue button labeled "OK" is positioned to the right of the password field. To the left of the "OK" button, the text "S'IDENTIFIER" is visible. The bottom of the screen shows the standard Android navigation bar with a square, a circle, and a triangle icon.

**Figure 23 Connexion**

Si non si le client n'est pas inscrit il peut inscrire de puis la si dessus, depuis l'insertion de ces information le nom, le prénom la date de naissance et le numéro de téléphone , nom d'utilisateur et le mot de passe.

The screenshot shows a mobile application interface for registration. At the top, there is a purple header bar with the text 'ClientRendez\_vous'. Below the header, the title 'Inscription' is centered. The form consists of several input fields with labels and values:

- Nom: medjber
- PreNom: wassila
- Date de naissance: 1994-06-04
- Telephone: 0796854223
- Nom d'utilisateur: medjber
- Mot de passe: 123123
- Confirmer le mot de passe: ..... (masked)

At the bottom of the form, there are two buttons: 'CONNECTION' in green text and 'OK' in white text on a purple background. The status bar at the top shows the time 20:55, signal strength, Wi-Fi, and battery level at 40%.

**Figure 24 inscription**

Et en fin en a la page rendez vous pour choisir et valider un rendez vous :

Le client choisi la date puis l'heure après il confirme son choix et et la le choix de modifie son rendez vous



Figure 25 réserver un rendez vous

### IV.4.3 Les tables de base de données :

- La table login :

Contient les noms d'utilisateur et sont mot de passe

#	ID_PATIENT	NOM_UTILISATEUR	PASS
1		2 wass	wass
2		3 medj	333
3		4 fares	222
4		5 abc	1234

- **LA table patiente :**

Contient les information de patients :Nom, prénom ,date de naissance et le numéro téléphone :

- **La table rendez-vous :**

#	ID	NOM	PRENOM	DATENAISSANCE	TELEPHON
1		2 madjbar	wassila	1998-06-30	0876464369
2		3 wass	sad	1998-06-30	0789654321
3		4 medjber	fares	2002-08-11	0987654321
4		5 med	med	1998-06-30	0789654321
5		6 asd	asd	1963-06-12	0789654321
6		7 mouhamed	mouham	1970-09-09	0789654321
7		8 medjber	saleh	1969-08-06	0789654321
8		9 abcd	abcd	1988-09-09	0789654321
9		10 medjber	samia	1966-08-09	0879654123
10		11 medjber	samia	1966-08-06	0987654321

Contient le rendez vous avec la date et l heure et le statut :

#	ID_RDV	ID_USER	DATE_RDV	HEURE	STATU
1		38	2 2021-09-13	11:00-11:30	att

## IV.5 Conclusion :

Ce chapitre a été, principalement, consacré le cote implémentation .par ailleurs nous avons présenté d'une part, les différents outils et environnements utilisés pour la réalisation de notre application. Et d'autre part nous avons donné une courte description des principales interfaces de l application.

---

## **Conclusion Générale :**

Notre Projet porte sur l'organisation et l'automatisation de la gestion d'une prise d'un rendez-vous, afin d'augmenter la fiabilité, l'efficacité de l'effort humain et de faciliter les tâches pénibles au sein d'un organisme ; en le déroulant sur une application android .

Le but principal de ce projet se base sur l'optimisation du temps ainsi les clients n'ont plus besoin d'attendre l'ouverture de structure pour prendre un rendez-vous. Nous les libérons de leurs contraintes professionnelles et familiales, d'où une simplicité de la charge quotidienne

Ce projet nous a permis d'avoir une approche complète du développement d'une application mobile à base de programmation android avec langage java.

Nous avons appliqué au maximum possible les règles de bases permettant d'avoir une application performante. Nous avons utilisé UML pour modéliser le système application gestion de rendez vous.

---

## Bibliographie

### [1] Mémoire online,

Samir Athmani ; «Protocole de sécurité pour les réseaux de capteurs sans fil ». Thèse de Magistère ; Université de Hadj Lakhdar-Batna ; Juillet 2010. (Consulté le 15/07/2021)

[2] [https\\_\\_\\_word-view.officeapps.live.com\\_wv\\_WordViewer\\_request](https___word-view.officeapps.live.com_wv_WordViewer_request).

(Consulté le 17/07/2021)

### [3] Mémoire online,

Belkheir Khaled et Haned Ahmed ; « Réseaux WiFi ad hoc ». Mémoire d'ingénieur ; Institut de télécommunication d'Oran ; Juin 2008. (Consulté le 17/07/2021)

### [4] Mémoire online,

Messaoud Belloula ; « La géolocalisation dans les réseaux de capteurs sans fil ; Etude de cas : Utilisation en agriculture ». Thèse de Magistère ; Université Hadj Lakhder-Batna. (Consulté le 05/08/2021)

[5] [http://projet.eu.org/pedago/sin/ISN/8-client\\_serveur.pdf](http://projet.eu.org/pedago/sin/ISN/8-client_serveur.pdf). (Consulté

le 01/09/2021)

[6] <https://www.javatpoint.com/socket-programming>

[7] <https://fr.wikipedia.org/wiki/Android>. (Consulté le 01/09/2021)

[8] <https://e-learning.univ-saida.dz/moodle/course/view.php?id=4604>

[9] Conception et réalisation d'une plate-forme d'enseignement à distance des bases de données, En vue de l'obtention du diplôme de licence en informatique Options : Sécurité informatique, Promotion : 2009/2010.

[10] <https://www.lucidchart.com/pages/fr/langage-uml>. (Consulté

le 10/09/2021)

[11] <https://fr.wikipedia.org/wiki/NetBeans>. (Consulté le 11/09/2021)

[12] [https://fr.wikipedia.org/wiki/Android\\_Studio](https://fr.wikipedia.org/wiki/Android_Studio). (Consulté le 11/09/2021)

[13] <https://fr.wikipedia.org/wiki/MySQL> / (Consulté le 11/09/2021)

**[14]** <https://gfx.developpez.com/tutoriel/java/network/>

(Consulté le 12/09/2021)

**[15]** <https://www.javatpoint.com/socket-programming>.

(Consulté le 15/09/2021)

**[16]** Les catégories des réseaux sans fil,

<https://www.clicours.com/les-categories-de-reseaux-sans-fils-la-technologie-wifi/> . (Consulté le 17/09/2021)

**[17]** des réseaux mobiles avec infrastructure

[https://www.memoireonline.com/05/12/5873/m\\_Etude-sur-les-protocoles-de-routage-dun-reseau-sans-fil-en-mode-Ad-Hoc-et-leurs-impacts-cas-de8.html](https://www.memoireonline.com/05/12/5873/m_Etude-sur-les-protocoles-de-routage-dun-reseau-sans-fil-en-mode-Ad-Hoc-et-leurs-impacts-cas-de8.html)

**[18]** Réseau en mode ad hoc

<https://theses.univ-oran1.dz/document/15201210t.pdf>.

(Consulté le 05/09/2021)

**[19]** architecture client/serveur et architecture p2p

[http://projet.eu.org/pedago/sin/ISN/8-client\\_serveur.pdf](http://projet.eu.org/pedago/sin/ISN/8-client_serveur.pdf).

(Consulté le 21/09/2021)

**[20]** Socket API

<http://csis.pace.edu/~marchese/CS865/Lectures/Liu4/sockets>. (Consulté

le 22/09/2021)

**[21]** logo android

<https://logos-world.net/android-logo/>

**[22]** Architecture d'Android, Mémoire online

[http://staff.univbatna2.dz/sites/default/files/bada\\_mosaab/files/chapitre1\\_architecture.pdf](http://staff.univbatna2.dz/sites/default/files/bada_mosaab/files/chapitre1_architecture.pdf).

**[23]** compilation et à l'exécution d'un programme Java standard

[http://pageperso.lif.univmrs.fr/~arnaud.labourel/programmation2/td\\_tp\\_projet/tp1.pdf](http://pageperso.lif.univmrs.fr/~arnaud.labourel/programmation2/td_tp_projet/tp1.pdf). (Consulté le 23/09/2021)

**[24]** cycle de vie d'une activité android

[http://www.iro.umontreal.ca/~dift1155/cours/ift1155/communs/Cours/2P/C02\\_CycledeVie\\_2P.pdf](http://www.iro.umontreal.ca/~dift1155/cours/ift1155/communs/Cours/2P/C02_CycledeVie_2P.pdf). (Consulté le 24/09/2021)

---

---

## Résumé

---

L'objectif de notre projet de fin d'étude pour l'obtention du diplôme de master en informatique option Réseaux informatiques et Système Répartis(RISR) consiste a concevoir et réaliser une application de gestion de rendez-vous a base d'android, cette application permettra au patient de réserver un rendez vous via un réseau sans fil, au serveur de base de donner d'un docteur.

---

## Abstract

---

The objective of our end-of-study project for obtaining the master's degree in computer science option Computer Networks and Distributed System (RISR) is to design and build an Android-based appointment management application; this application will allow the patient to book an appointment via a wireless network, to the base server to give from a doctor.