



MÉMOIRE

De fin d'études

En vue de l'obtention du diplôme de Master En Informatique

Option : Réseaux Mobilité et Systèmes Embarqués

THÈME

**Développement d'une application
Web - Android pour l'assistance
aux urgences médicales**

Proposé par : *Dr. S. Hamrioui*

Réalisé par :

HOCINI Yazid

et

MOUHEB Hakim



Remerciements

Nous avons l'honneur de réserver cette page pour témoigner notre reconnaissance à tous ceux qui ont contribué à ce travail.

Nous tenons, tout d'abord, à remercier vivement le Dr Sofiane HAMRIOUI, de m'avoir fait l'honneur de m'encadrer.

Nos remerciements s'adressent également à tous nos enseignants du Département Informatique, au personnel administratif sans oublier Mohand AZOUAOU.

Nous n'omettons pas de remercier nos camarades de la promotion.

Et enfin, nous adressons nos sincères remerciements aux membres du jury pour avoir accepté d'évaluer ce travail.

Dédicaces

A ma femme,

A ma fille Damia,

A la mémoire de ma mère.

A mon père, mes frères et mes sœurs,

A tous mes proches,

Et à tous mes amis.

YAZID.

A mon père,

A ma mère,

A mes frères et mes sœurs,

A tous mes proches,

Et à tous mes amis.

HAKIM.

Introduction générale	1
------------------------------------	----------

Chapitre I Généralités sur les réseaux

I.1 Introduction	4
I.2 Concept de réseau.....	4
I.3 Les réseaux filaires.....	4
I.3.1 Définition d'un réseau filaire	4
I.3.2 Médias de transport	5
I.3.3 Types de réseaux filaires	5
I.3.3.1 Les réseaux personnels (<i>Personal Area Network</i>)	5
I.3.3.2 Les réseaux locaux (<i>Local Area Network</i>).....	5
I.3.3.3 Les réseaux métropolitains (<i>Metropolitan Area Network</i>).....	6
I.3.3.5 Limites des réseaux filaires	6
I.4 Les réseaux sans fil	6
I.4.1 Définition d'un réseau sans fil	6
I.4.2 Classification des réseaux sans fil.....	7
I.4.2.1 Classification suivant la distance séparant les terminaux mobiles.....	7
A- Les réseaux personnels sans fil (<i>Wireless Personal Network</i>).....	8
B- Le réseau local sans fil (WLAN).....	8
C- Le réseau sans fil métropolitain (WMAN).....	8
D- Le réseau sans fil WWAN (<i>Wireless Wide Area Network</i>).....	8
E- Générations de la technologie mobile.....	8
I.4.2.2 Classification suivant l'infrastructure	9
A- Les réseaux sans fil avec infrastructure.....	9
B- Les réseaux sans fil sans infrastructure (ad hoc)	9
I.4.3 Définition d'un réseau ad hoc mobile	10
I.4.3.3 Modélisation d'un réseau Ad hoc Mobile.....	10
I.4.3.4 Topologie d'un réseau mobile ad hoc	11
I.4.3.5 Protocoles de routage dans les réseaux ad hoc mobiles	11
I.4.3.6 Les protocoles de routage proactifs.....	12
I.4.3.6.1 Le protocole OLSR (<i>Optimized Link-State Routing</i>)	12
I.4.3.6.2 Le protocole DSDV (<i>Destination-Sequenced Distance Vector routing</i>)	13
I.4.3.7 Les protocoles de routage réactifs.....	14
I.4.3.7.1 Le protocole AODV (<i>Ad Hoc On-demand Distance Vector</i>)	14

I.4.3.8 Les protocoles hybrides.....	15
I.4.3.8.1 Le protocole ZRP (<i>Zone Routing Protocol</i>).....	15
I.4.3.8.2 Le protocole CBRP	16
I.4.3.9 Les caractéristiques des réseaux ad hoc mobiles	16
I.5 Conclusion.....	17

Chapitre II Géolocalisation

II.1 Introduction.....	19
II.2 Définition de la géolocalisation	19
II.3 Techniques de géolocalisation	20
II.3.1 Géolocalisation par satellites	20
II.3.2 Géolocalisation par GPRS/GSM	21
II.3.3 Géolocalisation par Wi-Fi	21
II.3.4 Géolocalisation par adresse IP	22
II.3.5 Géolocalisation par RFID	22
II.4 Combinaison des différentes techniques.....	23
II.5 Plateformes logicielles de géolocalisation	24
II.5.1 Composants.....	24
II.5.2 Architecture temps réel	24
II.6 Types de terminaux existants.....	25
II.6.1 Modes de fonctionnement.....	25
II.6.2 Modes de communication.....	27
II.7 Conclusion	28

Chapitre III Analyse et conception

III.1 Introduction	30
III.2 Problématique.....	30
III.3 Rappels sur l'UML	31
III.3.1 Définition.....	31
III.3.2 Le formalisme d'UML	31
III.3.3 Définition architecture avec UML.....	32
III.3.4 Les Diagrammes d'UML.....	33

III.4 Analyse	34
III.4.1 Spécification des besoins.....	34
III.4.2. Architecture générale du système.....	34
III.4.3 Identification des acteurs	37
III.4.4 Les cas d'utilisation	37
III.4.5 Diagramme de cas d'utilisation	38
III.4.6 Identification des scénarios	40
III.4.6.1 Les scénarios par taches d'un réceptionniste.....	41
III.4.6.2 Les scénarios par taches d'un médecin.....	42
III.4.6.3 Les scénarios par taches d'un administrateur	43
III.5. Conception.....	45
III.5.1 Diagramme de séquence.....	45
III.5.2 Diagramme des classes.....	48
III.5.2.1 Les règles de gestion.....	49
III.5.3 Conception de la base de données	51
III.6 Conclusion.....	52

Chapitre IV Réalisation

IV.1 Introduction	54
IV.2 Outils utilisés.....	54
IV.3 Android.....	56
IV.3.1 Définition	56
IV.3.2 Le kit de développement (SDK) d'Android	56
IV.3.3 Composantes d'une application Android	56
IV.3.4 Cycle de vie d'une application Android.....	58
IV.4 Description des interfaces de l'application Server	60
IV.4.1 Description de l'interface d'authentification.....	60
IV.4.2 Description de l'interface principale	61
IV.4.3 Description de l'interface de l'administrateur.....	62
IV.4.3.1 Formulaire d'ajout de médecin.....	63
IV.4.3.2 Interface de gestion des services	64
IV.4.3.3 Formulaire d'ajout de service.....	64
IV.5 Description des interfaces de l'application Client (Android).....	65

IV.5.1 Description de l'interface d'authentification.....	65
IV.5.2 Description de l'interface disponibilité	66
IV.5.3 Description de l'interface de réception d'alerte	67
IV.6 Conclusion.....	68
Conclusion générale	70

Introduction générale

Avec l'accroissement démographique en nette progression et le vieillissement d'une bonne frange de la population, les hôpitaux ne peuvent plus répondre efficacement aux sollicitations accrues des patients. La tension s'accroît sur ces lieux dans les grandes villes particulièrement en période estivale où il y a une forte mobilité des personnes occasionnant beaucoup d'accidents et plusieurs autres maladies telles que les intoxications alimentaires. D'où la gestion des urgences devient un problème épineux et quasi ingérable.

L'évolution des technologies de l'information, le développement incessant et continu des moyens de communication et le penchant de plus en plus fort vers l'utilisation des appareils sans fils, ont fait émerger les réseaux téléphoniques mobiles et par la suite le développement de diverses fonctionnalités et services de ces derniers. Ceci a induit à la généralisation de l'usage du téléphone mobile par la majeure partie de la population dont fait partie le personnel médical.

C'est pour profiter de cette généralisation du téléphone mobile et remédier aux problèmes rencontrés quotidiennement aux niveaux des services des urgences des hôpitaux que l'idée de notre projet est née.

Notre travail consiste donc à développer une application de gestion des urgences médicales qui s'intéressera à mieux gérer l'élément clé pour la prise en charge des malades, en l'occurrence le médecin.

Le médecin, possédant certaines caractéristiques de spécialité et de disponibilité, est une ressource dont dépendra la réponse à l'urgence médicale signalée. La problématique est celle de pouvoir joindre, à tout moment, cette ressource en effectuant une recherche de personnel répondant aux critères spécifiques tels que la spécialité et la localisation.

Pour ce faire, nous avons subdivisé notre travail en quatre chapitres :

Le premier chapitre nous l'avons consacré aux généralités sur les réseaux ;

Dans le deuxième chapitre nous avons décrit les différentes techniques de géolocalisation.

Le troisième chapitre décrit l'analyse et la conception de notre système à travers son architecture globale en définissant les différentes fonctionnalités attendues du système en suivant une modélisation en langage UML.

Le quatrième chapitre est consacré à la présentation de la réalisation de l'application. Nous commençons par décrire les outils de développement utilisés puis les différentes interfaces et une brève explication pour chacune d'elle pour donner une vue générale et complète sur le système.

Enfin, nous terminons par une conclusion générale résumant l'apport de notre travail et des perspectives.

Chapitre I

Généralités sur les réseaux

I.1 Introduction

Les réseaux informatiques, de par leurs diverses utilités, sont devenus incontournables aujourd'hui. On les retrouve dans les entreprises, quelque soient leurs tailles, et même chez les particuliers. Ils permettent de mettre en œuvre des applications très diverses, d'un simple partage de connexion Internet aux applications les plus sophistiquées.

Dans ce premier chapitre nous présenterons les différents types de réseaux répartis leurs types de liaisons ainsi que sur leurs étendues. Nous mettrons l'accent sur les réseaux mobiles qui sont au cœur de la thématique de notre travail.

I.2 Concept de réseau

Un réseau est un ensemble d'objets interconnectés. Il permet de faire circuler des éléments entre chacun de ces objets selon des règles bien définies. Selon le type d'objets, on parlera parfois de :

- Réseau de transport : ensemble d'infrastructures et de disposition permettant de transporter des personnes et des biens entre plusieurs zones géographiques.
- Réseau téléphonique : infrastructure permettant de faire circuler la voix entre plusieurs postes téléphoniques.
- Réseau de neurones : ensemble de cellules interconnectées entre elles.
- Réseau informatique : ensemble d'ordinateurs reliés entre eux grâce à des lignes physiques et échangeant des informations sous forme de données numériques.[1]

I.3 Les réseaux filaires

I.3.1 Définition d'un réseau filaire

Le réseau filaire utilise des câbles (Ethernet, coaxiaux, fibre optique) pour relier des ordinateurs et des périphériques via des concentrateurs ou des commutateurs suivant une topologie donnée.

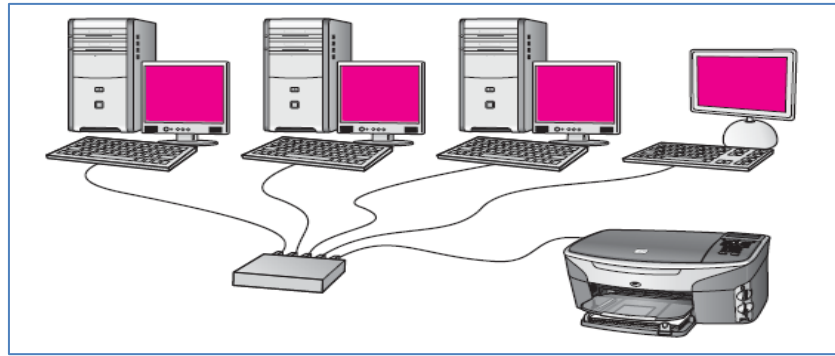


Figure I.1 Exemple de réseau filaire

I.3.2 Médias de transport

Les supports utilisés pour l'interconnexion des équipements dans les réseaux filaires sont deux types :

- Les supports métalliques : Ils reposent sur la propriété de la conductivité électrique. Ils sont les câbles Ethernet qui sont constitués de 4 paires de fils torsadés et les câbles coaxiaux.
- Fibre optique : la transmission des données se fait par propagation d'un rayon lumineux

I.3.3 Types de réseaux filaires

Selon le nombre de machines formant le réseau et la manière d'utilisation de celui-ci, les réseaux locaux sont divisés en quatre types :

I.3.3.1 Les réseaux personnels (*Personal Area Network*)

Ces réseaux sont constitués d'un nombre restreint d'équipements informatiques habituellement utilisés dans le cadre d'un usage personnel.

I.3.3.2 Les réseaux locaux (*Local Area Network*)

Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie dont la plus répandue étant Ethernet¹

¹ Ethernet (aussi connu sous le nom de norme IEEE 802.3) est un standard de transmission de données pour réseau local basé sur le principe suivant : Toutes les machines du réseau Ethernet sont connectées à une même ligne de communication, constituée de câbles cylindriques.

I.3.3.3 Les réseaux métropolitains (*Metropolitan Area Network*)

Un réseau métropolitain est un réseau couvrant une ville où plusieurs lieux sont interconnectés. Les MAN (*Metropolitan Area Network*) interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux nœuds distants de communiquer comme si ils faisaient partie d'un même réseau local.[2]

I.3.3.4 Les réseaux longue distance (*Wide Area Network*)

Un WAN (*Wide Area Network* ou réseau étendu) interconnecte plusieurs LANs à travers de grandes distances géographiques. Les infrastructures physiques peuvent être terrestres, sous-marines, spatiales ou satellitaires.

Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau.

Le plus connu des WAN est le réseau Internet.

I.3.3.5 Limites des réseaux filaires

Malgré les avantages de sécurité et de stabilité qu'offrent les réseaux filaires par rapport aux réseaux sans fils, certaines limites les rendent encombrants et onéreux. Nous citons :

- Le coût élevé de l'installation de ce type de réseaux comparativement aux réseaux sans fil
- L'absence de mobilité et de liberté de mouvement.
- Perte de câble, de goulotte et autres prises murales en cas de déplacement.

I.4 Les réseaux sans fil

I.4.1 Définition d'un réseau sans fil

Le concept de réseaux sans fil est étroitement associé au support de transmission. Un système est dit sans fil s'il propose un service de communication totalement indépendant de prises murales et de câbles. Dans cette configuration d'autres moyens d'accès sont exploités, tels que l'infrarouge et les ondes hertziennes.[3]

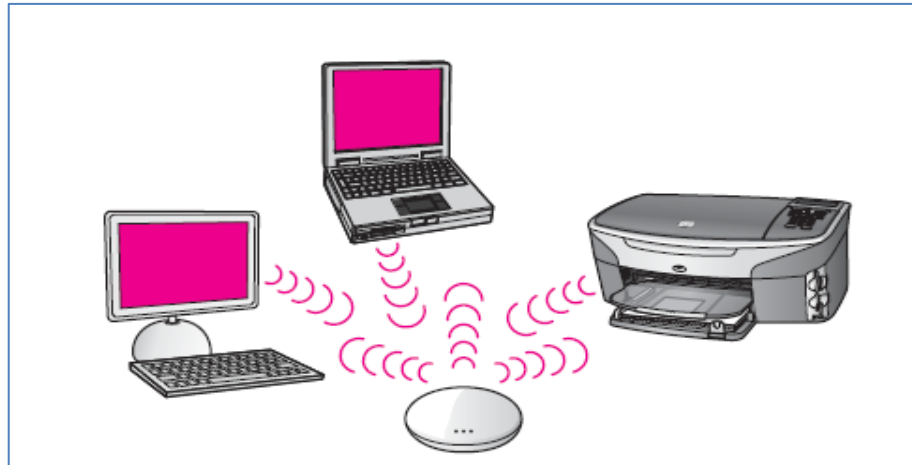


Figure I.2 Exemple de réseau sans fil

I.4.2 Classification des réseaux sans fil

I.4.2.1 Classification suivant la distance séparant les terminaux mobiles

La figure 3 décrit les différentes catégories de réseaux définies en fonction de la taille et de la zone de couverture.

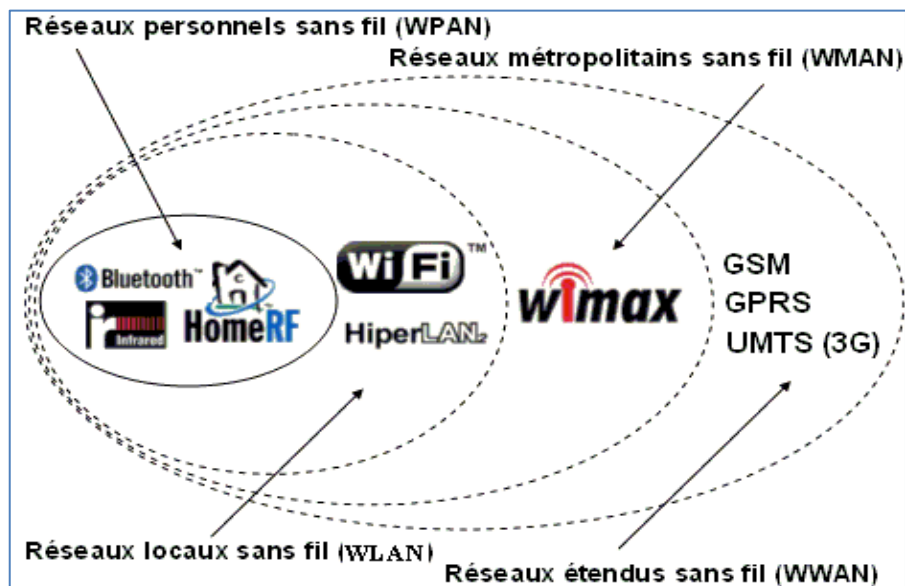


Figure I.3 Types de réseaux sans fil

A- Les réseaux personnels sans fil (*Wireless Personal Network*)

Ce sont des réseaux sans fil d'une faible portée (quelques mètres). Ce type de réseau sert généralement à relier une imprimante, un téléphone portable et des appareils domestiques à un ordinateur. Parmi les technologies WPAN, on peut citer Bluetooth² et ZigBee³.

B- Le réseau local sans fil (WLAN)

Les réseaux locaux sans fil couvrent des zones sur des dizaines à centaines de mètres. Ils interconnectent les équipements informatiques d'une même entreprise ou d'une même université.

C- Le réseau sans fil métropolitain (WMAN)

Le réseau métropolitain sans fil visent à remplacer les modems ADSL, que l'on trouve sur les réseaux téléphoniques fixes, pour donner à des utilisateurs finaux des débits importants pour du hertzien, jusqu'à plusieurs mégabits par seconde.

D- Le réseau sans fil WWAN (*Wireless Wide Area Network*)

Le réseau étendu sans fil (WWAN) est également connu sous le nom du réseau cellulaire mobile. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil.

E- Générations de la technologie mobile

La technologie mobile est classée en générations :

- La 1^{ère} génération (1G) supportait uniquement le trafic vocal.
- La 2^{ème} génération de réseaux cellulaires (2G) permet la transmission de la voix ainsi que des données numériques de faibles volumes tels que les messages texte SMS. Le GSM (Global System for Mobile) permet un débit maximal de 9.6 Kbits/s, est l'un des principaux standard de la téléphonie mobile 2G.
- La 3^{ème} génération (3G) désigne une génération de normes de téléphonie mobile. Elle permet des débits (de 2 à 42 Mb/s).

² Bluetooth est un standard de communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF.

³ ZigBee est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur la norme IEEE 802.15.4

UHF : Bande des Ultra hautes fréquences est la bande du spectre radioélectrique comprise entre 300 MHz et 3 000 MHz.

- La 4^{ème} génération (4G) permet d'atteindre le « très haut débit mobile », c'est-à-dire des transmissions de données à des débits théoriques supérieurs à 100 Mb/s, voire supérieurs à 1 Gb/s.

I.4.2.2 Classification suivant l'infrastructure

A- Les réseaux sans fil avec infrastructure

Dans ce mode de fonctionnement, le réseau est composé d'un point d'accès appelé Station de Base, muni d'une interface de communication sans fil pour la communication directe avec les sites ou unités mobiles. Une station de base couvre une station géographique limitée dite portée.

B- Les réseaux sans fil sans infrastructure (ad hoc)

Un réseau ad hoc mobile est une collection d'entités mobiles, appelées stations ou nœuds, interconnectées par une technologie sans fil.

L'infrastructure n'est composée que de stations elles-mêmes qui jouent, à la fois, le rôle de terminaux et de routeurs pour permettre le passage de l'information d'un terminal vers un autre.

Les réseaux mobiles ad hoc sont généralement mis en œuvre lorsqu'une infrastructure sans fil est difficilement envisageable ou indisponible. De même, lors de l'utilisation de courte durée ou soudaine de certaines applications. Il est aisé de recourir à ce type de réseaux car il est facilement déployable et auto-organisé. Son faible coût participe aussi à son succès [4].

Les MANETs (*Mobile Ad-hoc NETWORKs*) sont utiles dans de nombreux domaines. Nous citons le domaine militaire en zone de combats, la coordination des opérations de secours lors des catastrophes naturelles et dans les missions d'exploration.

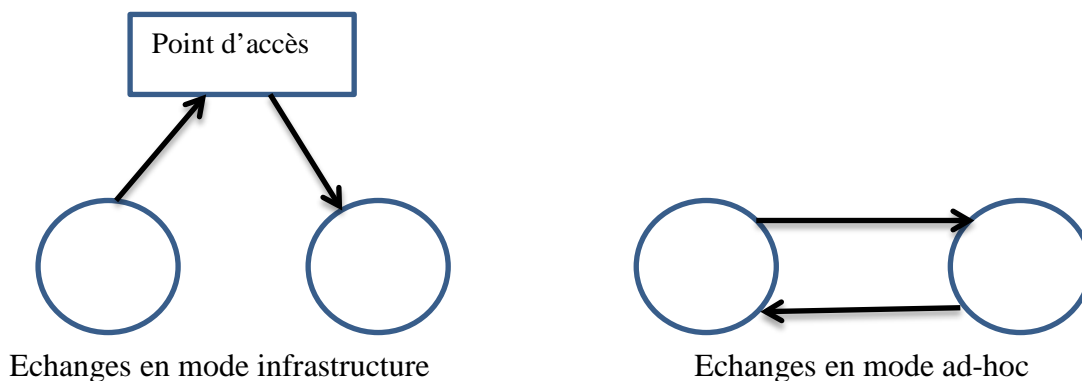


Figure I.4 Modes d'échange dans les réseaux sans fil

L'utilisation la plus simple et la plus courante des réseaux ad-hoc est faite par les réseaux sans fil Wi-Fi en permettant une mise en place rapide d'une connexion réseau entre deux ordinateurs.

I.4.3 Définition d'un réseau ad hoc mobile

Un réseau ad-hoc mobile est un réseau créé par une réunion spontanée de mobiles capables de communiquer grâce à leurs interfaces sans fils, sans aucune infrastructure ni configuration préalable. Les nœuds d'un tel réseau coopèrent pour la gestion du réseau. Un MANET se caractérise par sa méthode de découverte de nœuds, son algorithme de routage, et le modèle de mobilité régissant le mouvement des mobiles [5].

I.4.3.3 Modélisation d'un réseau Ad hoc Mobile

Un réseau mobile ad hoc consiste en une grande population, relativement dense, d'unités mobiles qui se déplacent dans un territoire quelconque. Il peut être modélisé par un graphe $G_t = (V_t, E_t)$ où V_t représente l'ensemble des nœuds (les unités mobiles) du réseau et E_t modélise l'ensemble des connexions qui existent entre ces nœuds [6].

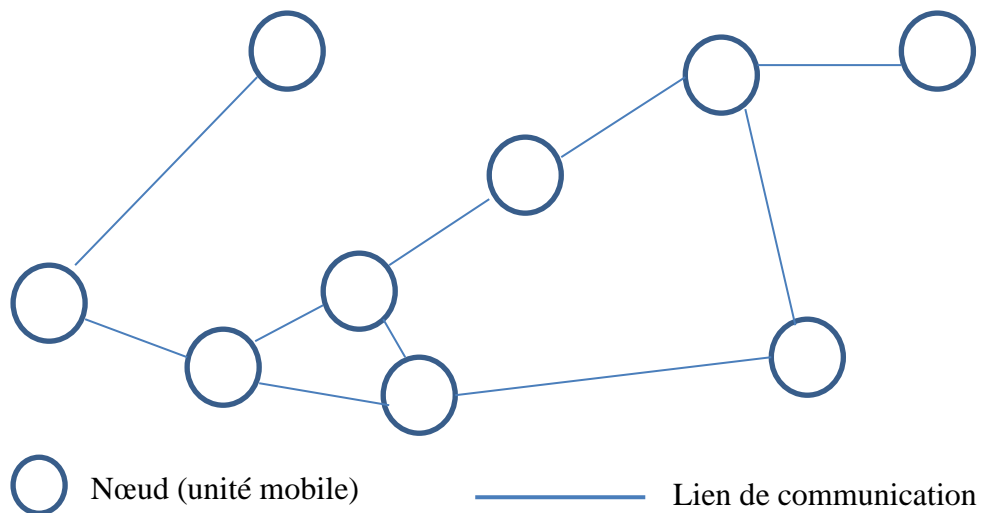


Figure I.5 Modélisation d'un réseau ad hoc

I.4.3.4 Topologie d'un réseau mobile ad hoc

Etant donné que les unités mobiles sont en mouvement, la topologie du réseau peut changer à tout moment. Elle est donc dynamique et imprévisible ce qui engendre des déconnexions fréquentes des unités mobiles.

La figure suivante montre un exemple de changement de topologie dans un réseau où les unités sont mobiles.

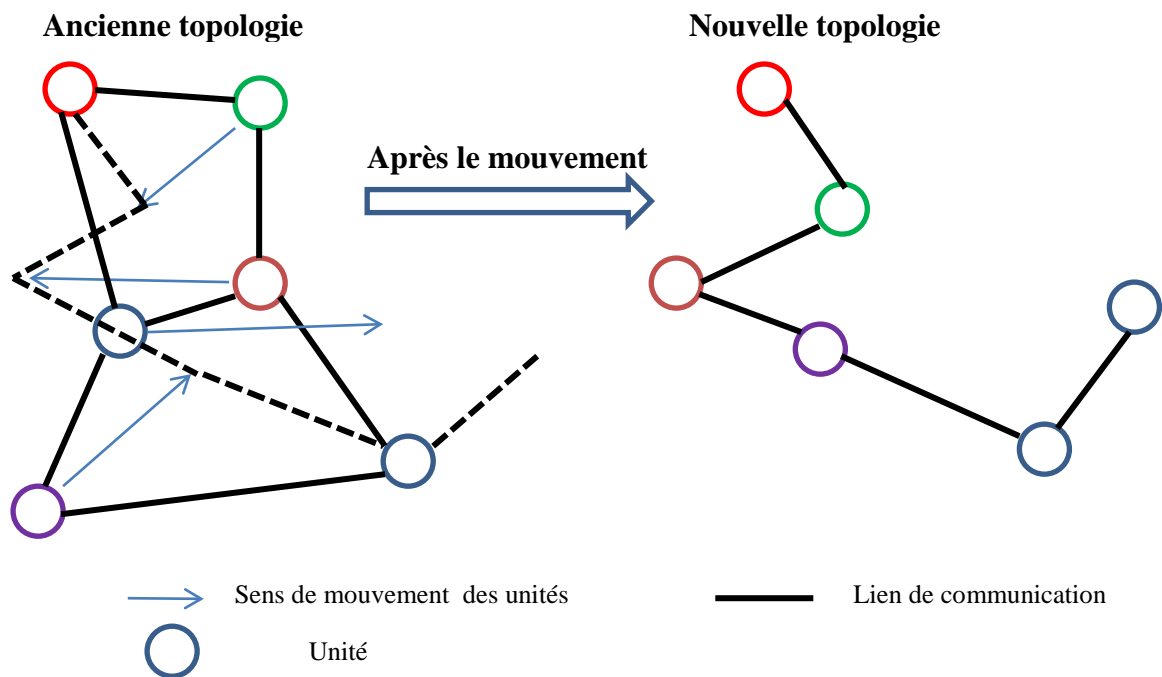


Figure I.6 Changement de la topologie des réseaux ad hoc mobiles

I.4.3.5 Protocoles de routage dans les réseaux ad hoc mobiles

Le routage est la méthode d'acheminement des informations d'une source vers une destination à travers un réseau donné. Le problème de routage consiste à déterminer une transmission optimale des paquets par rapport à certains critères de performances.

Dans les réseaux ad hoc, il s'agit de trouver une méthode d'acheminement pour un grand nombre de nœuds dans un environnement caractérisé par des changements rapides de la topologie du réseau dus à la mobilité des hôtes, ainsi que d'autres caractéristiques comme l'absence de l'infrastructure, la limitation de la bande passante, la limitation de source d'énergie et les modestes capacités de traitements et de mémoire [7].

Dans les réseaux ad hoc mobiles, plusieurs protocoles de routage existent. Ceux-ci, sont classés en trois familles : les proactifs, les réactifs et les hybrides.

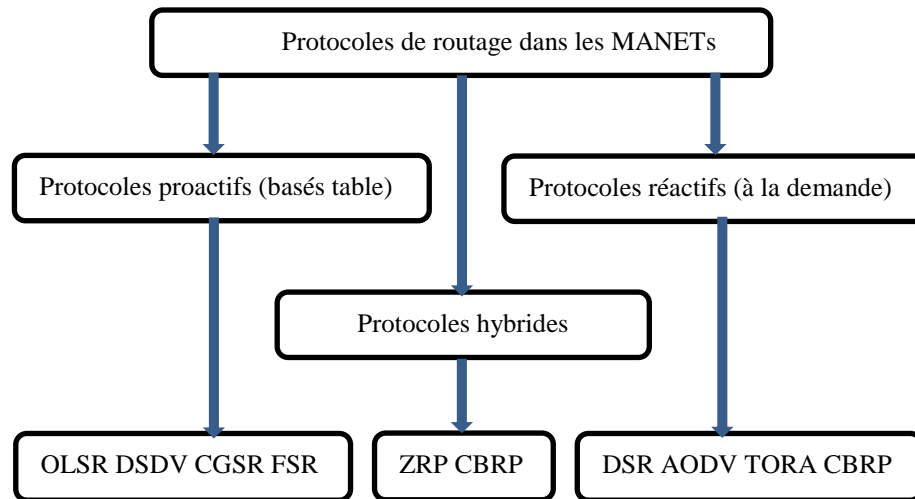


Figure I.7 Classification des familles de protocoles

I.4.3.6 Les protocoles de routage proactifs

Les unités mobiles cherchent à maintenir une table de routage cohérente, même en l'absence de communication.

Chaque nœud construit une table de routage par la découverte d'autres nœuds dans tout le réseau, parmi ces protocoles OLSR et DSDV.

I.4.3.6.1 Le protocole OLSR (*Optimized Link-State Routing*)

Le protocole OLSR est proposé par des chercheurs de l'INRIA⁴ et fait l'objet de travaux de standardisation depuis 2003 au sein de la IETF⁵ dans le groupe MANET. C'est un protocole proactif basé sur une approche modifiée, dont la version originale ou chaque nœud propage des informations sur l'état des liens à tous les nœuds du réseau. L'altération apportée par OLSR consiste à réduire l'overhead⁶ en ne diffusant cette information que par quelques nœuds du réseau.

⁴ Institut national de recherche en informatique et en automatique, France

⁵ Internet Engineering Task Force : grande communauté internationale ouverte de concepteurs de réseaux

⁶ Surcharge du réseau

Dans ce protocole proactif, chaque nœud transmet sa liste de voisins. De cette manière chaque nœud est au courant des nœuds qui circulent dans son voisinage à 2 sauts. Il utilise un algorithme de sélection de relais multipoints MPR (*Multi-Point Relay*). L'ensemble des relais d'un nœud N donné est construit à partir de l'ensemble minimal des voisins à un saut de N. De telle sorte que chaque voisin à deux sauts de N ait au moins un même MPR. Chaque nœud dans OLSR choisit indépendamment ses relais multi-points. La seule connaissance dont il a besoin est celle relative à ses voisins à deux sauts. Quand un nœud diffuse un message, tous ses voisins le reçoivent. Les MPR n'ayant pas reçu ce message auparavant le rediffusent. Donc, l'overhead causé par l'inondation diminue considérablement dans OLSR. Le nœud diffuse périodiquement l'information sur l'état des liens avec leurs MPRs dans le réseau. La périodicité de l'envoi de ces messages est sujette à la détection du changement de l'ensemble des MPRs.

I.4.3.6.2 Le protocole DSDV (*Destination-Sequenced Distance Vector routing*)

L'algorithme de routage appelé "Vecteur de Distance à Destination Dynamique Séquencée" ou DSDV a été conçu spécialement pour les réseaux mobiles. Il est basé sur l'idée classique de l'algorithme distribué de Bellman-Ford, DBF, en y apportant des solutions à ses éventuels problèmes. Ce protocole fonctionne de façon suivante [8] :

Chaque nœud maintient une table de routage qui contient :

- Toutes les destinations possibles.
- Le nombre de saut nécessaire pour atteindre la destination.
- Le numéro de séquence qui correspond à un nœud destination.

Afin de maintenir la consistance des tables de routage dans une topologie qui varie rapidement, chaque nœud du réseau transmet périodiquement sa table de routage à ses voisins directs. Le nœud peut aussi transmettre sa table de routage si le contenu de cette dernière subit des changements significatifs par rapport au dernier contenu envoyé.

La mise à jour doit permettre à un nœud de pouvoir localiser, dans la plupart des cas, une autre unité du réseau.

La mise à jour peut être complète ou incrémentale.

Dans la mise à jour complète, le nœud transmet la totalité de la table de routage aux voisins. Ce qui nécessite l'envoi de plusieurs paquets de données ; alors que dans une mise à jour incrémentale, juste les entrées qui ont subi un changement par rapport à la dernière mise à jour, sont envoyées ce qui réduit le nombre de paquets transmis.

I.4.3.7 Les protocoles de routage réactifs

Cette approche propose de ne construire une route vers la destination que si un nœud a besoin de communiquer. Ceci permet de résister à la mobilité mais génère une surcharge supplémentaire [5].

Lorsque le réseau a besoin d'une route, une procédure de découverte globale de routes est lancée, et cela dans le but d'obtenir une information spécifique, inconnue au préalable. Plusieurs approches peuvent être appliquées dans la découverte des routes.

La majorité des algorithmes utilisés sont basés sur le mécanisme d'apprentissage en arrière (*backward learning*). Le nœud source, qui est à la recherche d'un chemin vers la destination, diffuse par inondation une requête dans le réseau. Lors de la réception de la requête, les nœuds intermédiaires essaient de faire apprendre le chemin au nœud source et de sauvegarder la route dans la table envoyée. Une fois la destination atteinte, elle peut envoyer une réponse en utilisant le chemin tracé par la requête, un chemin full duplex est alors établi entre le nœud source et le nœud destination.

I.4.3.7.1 Le protocole AODV (*Ad Hoc On-demand Distance Vector*)

Dans la famille des routages à la demande, le protocole AODV est l'un des plus célèbres. Dans cette approche, quand un nœud source S a des données à envoyer au nœud destination D, mais ne possède pas tout le cheminement vers celle-ci, il doit lancer une Route REQuest (RREQ). Ce paquet est alors envoyé à travers le réseau. Chaque nœud recevant une RREQ, la fait suivre à ses voisins, s'il ne l'a pas déjà fait au moins une fois, ou s'il n'est pas le nœud destination de ce message, et sous la condition que le compteur Time to Live (TTL) ne soit pas à zéro [5].

Le paquet RREQ comporte l'identifiant de la source (SrcID), l'identifiant de la destination (DestID), numéro de séquence de la source (SrcSeqNum), numéro de séquence de la destination (DesSeqNum), l'identification de la diffusion (BcastID), et le TTL. Le DesSe-

qNum indique la fraîcheur du chemin accepté par la source. Lorsqu'un nœud reçoit un paquet RREQ, et s'il en est le destinataire ou connaît un chemin valide jusqu'au destinataire, il renvoie à la source du message une réponse Route REPlY (RREP). Sinon, il retransmet le RREQ à ses voisins.

Le nœud intermédiaire est en mesure de déterminer si un cheminement est valide ou non, en comparant son propre numéro de séquence avec le DesSeqNum dans le RREQ. Un nœud intermédiaire est en mesure de savoir si le RREQ est déjà traité ou non, en comparant le BcastID au SrcID. Lors de la retransmission du RREQ, chaque nœud intermédiaire y injecte l'adresse du nœud le précédant et son BcastID.

Un timer est utilisé pour supprimer cette entrée dans le cas où un RREP paquet n'est pas reçue avant sa date d'expiration. Lors de la réception d'un paquet RREP, un nœud intermédiaire mémorise aussi des informations sur le nœud à partir duquel ce paquet est reçu. Ce nœud étant le saut suivant lors de la transmission des données au nœud destination qui a initié le RREP.

I.4.3.8 Les protocoles hybrides

Il résulte d'une combinaison des avantages des deux familles précédentes, ce qui offre un bon compromis entre persistance de la route et surcharge du réseau. En général, les propositions hybrides exploitent la hiérarchisation du réseau en utilisant des protocoles réactifs et proactifs et sont alors intégrés à différents niveaux de la hiérarchie.

I.4.3.8.1 Le protocole ZRP (*Zone Routing Protocol*)

Ce protocole combine les deux approches, proactives et réactives, pour en tirer le maximum d'avantages et combler ainsi les désavantages de chacune par les points forts de l'autre. Il consiste en la division du réseau, en fonction de la distance entre les nœuds, en plusieurs parties appelées zones de routage. ZRP utilise deux approches proactive (IARP) pour le routage dans la même zone (intera-zone routing protocol) et réactive (IERP) entre les zones (inter-zone routing protocol). L'IARP maintient une information sur l'état de lien des nœuds à une distance d . Dans le cas où le nœud source et la destination sont dans la même zone, la route est alors disponible immédiatement. La plupart des protocoles proactifs existants pourraient être utilisés en tant que IARP dans ZRP. Lorsque la source et la destination appartiennent à deux zones différentes, l'IERP effectue une découverte de route de la même manière que c'est dans les protocoles réactifs, sauf qu'elle passe par les nœuds périphériques.

I.4.3.8.2 Le protocole CBRP

Le protocole CBRP (*Cluster Based Routing Protocol*) utilise un mécanisme hiérarchique à deux niveaux. Il définit la notion de cluster de nœuds : c'est un groupe de nœuds formé de membres et d'un chef de cluster (*clusterhead*), chaque membre du cluster étant à portée radio directe du clusterhead. Les clusterheads doivent maintenir une connectivité entre eux, ce qui assure la connectivité de tout le réseau. Les nœuds du réseau passent par les clusterheads pour la transmission de leurs données à travers le réseau [9].

I.4.3.9 Les caractéristiques des réseaux ad hoc mobiles

Les réseaux ad hoc mobiles sont caractérisés par :

1. Une topologie dynamique: Les unités mobiles du réseau, se déplacent d'une façon libre et arbitraire. Par conséquent la topologie du réseau peut changer, à des instants imprévisibles, d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unis ou bidirectionnels.
2. Une bande passante limitée : Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste.
3. Des contraintes d'énergie : Les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries ou les autres sources consommables. Le paramètre d'énergie doit être pris en considération dans tout contrôle fait par le système.
4. Une sécurité physique limitée : Les réseaux mobiles ad hoc sont plus touchés par le paramètre de sécurité, que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.
5. L'absence d'infrastructure : Les réseaux ad hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructures préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.
6. Auto-organisé et distribué : Les réseaux MANETs ne disposent pas de point central pour coordonner ou centraliser les échanges. De fait, ces réseaux doivent s'auto-organiser afin d'opérer. De plus, l'absence de centralisation demande à chaque acteur du réseau de participer au bon fonctionnement du réseau (distribution)[4].

7. Multi-saut : Comme la portée des stations est limitée, il peut s'avérer nécessaire que des stations agissent en tant que pont intermédiaire pour transmettre un paquet d'une source vers une destination. Par conséquent, les nœuds d'un réseau MANET agissent en tant que routeur et relayent les paquets qu'ils reçoivent pour participer au routage multi-saut.

I.5 Conclusion

Dans ce chapitre, nous avons introduit les concepts des réseaux informatiques. Nous avons décrit leurs différents types ainsi que leurs caractéristiques et certains protocoles de communication.

Chapitre II :

Géolocalisation

II.1 Introduction

Nous abordons dans ce chapitre la géolocalisation ainsi que les différentes techniques de localisation tout en mettant l'accent sur la technique choisie dans ce travail.

II.2 Définition de la géolocalisation

C'est un procédé permettant de positionner un objet sur un plan ou une carte à l'aide de ses coordonnées géographiques. Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un système de positionnement par satellites ou par à d'autres techniques et de publier, en temps réel ou de façon différée, ses coordonnées géographiques¹. [10]

La géolocalisation c'est déterminer la position précise d'un individu dans un environnement bien déterminé.

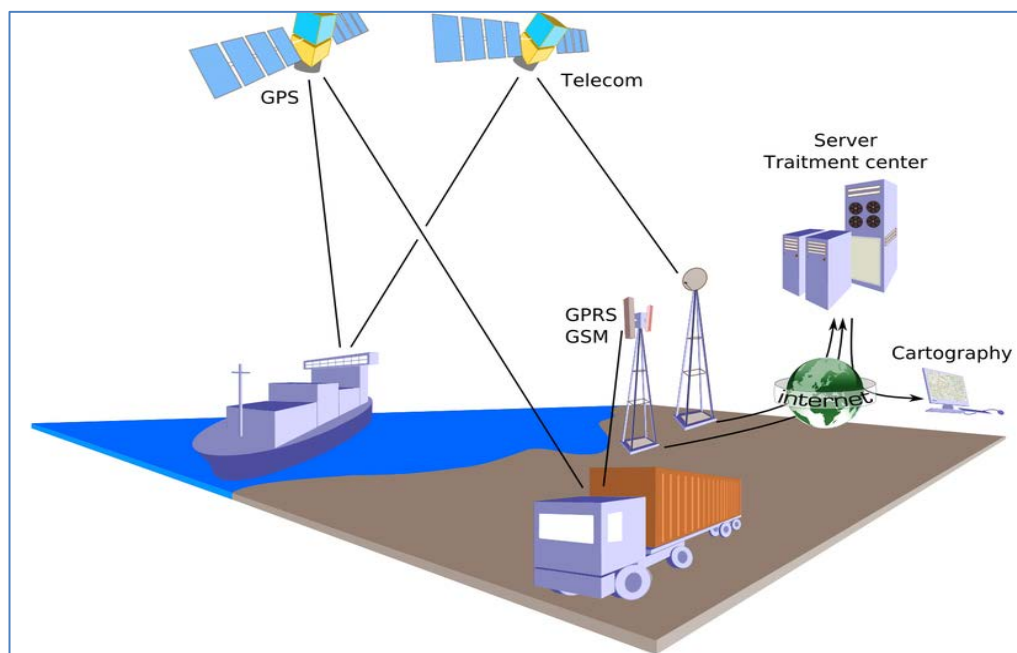


Figure II.1 Différents modes de géolocalisation [10]

¹ Coordonnées géographiques : Système de trois coordonnées qui sont le plus souvent : la latitude, la longitude et l'altitude (ou l'élévation) par rapport au niveau de la mer en un point arbitrairement choisi.

Latitude : représentée par une valeur angulaire, expression de la position d'un point sur Terre (ou sur une autre planète), au nord ou au sud de l'équateur qui est le plan de référence

Longitude : représentée par une valeur angulaire, expression du positionnement est-ouest d'un point sur Terre (ou sur une autre planète). La longitude de référence sur Terre est le méridien de Greenwich.

Altitude : est l'élévation verticale d'un lieu ou d'un objet par rapport à un niveau de base. C'est une des composantes géographique et biogéographique qui explique la répartition de la vie sur terre.

II.3 Techniques de géolocalisation

Les hommes ont toujours eu besoin de localiser les objets et de se situer dans l'environnement. Pour répondre à cette nécessité, plusieurs techniques ont été utilisées, des pierres, des montagnes, des fleuves jusqu'aux corps célestes. Depuis l'avènement de la technologie, l'homme a inventé d'autres techniques qui lui permettent de se localiser dans son environnement.

Les systèmes de géolocalisation permettent aux utilisateurs de terminaux de communications électroniques d'obtenir des services personnalisés qui prennent en compte leur situation géographique (informations météorologiques, navigation routières, balise de détresse, surveillance policière...).

Les informations relatives à la géolocalisation de l'utilisateur peuvent être recueillies par différents procédés que nous expliquerons dans ce qui suit.

II.3.1 Géolocalisation par satellites

La géolocalisation par satellites consiste à calculer la position actuelle d'un appareil mobile sur la face terrestre. Il existe 3 services mondiaux de positionnement par satellites, le GPS, GLONASS et GALILEO. GLONASS est un dispositif militaire Russe qui n'est plus entretenu.[11]

Le réseau satellite de positionnement le plus connu est le GPS (*Global Positionning System*)[12], bien que l'alternative Européenne nommée GALILEO [13] soit en cours de développement.

Grâce à une vingtaine de satellites répartis sur 6 orbites autour du globe, cette technique utilisée par GPS permet une localisation très précise (de 10 à 20 mètres) de l'objet qui peut alors être représenté physiquement sur une carte. Toutefois dès lors que l'objet est à l'intérieur d'un bâtiment, la localisation n'est plus possible car l'appareil ne reçoit plus les ondes émises par le satellite.

Le principe de localisation GPS repose sur l'émission de signaux codés véhiculés par une onde porteuse, selon deux modes de fonctionnement :

- Un mode précis de positionnement réservé à des utilisateurs identifiés (code P) ; une précision de l'ordre de 10 m
- Un mode standard de positionnement, sans restriction d'utilisation (code C/A) ; précision de l'ordre de 100 m.

Un récepteur reçoit donc simultanément les signaux codés en provenance de plusieurs satellites situés à des distances différentes du lieu d'observation. Le décodage de ces signaux permet d'évaluer ces distances et d'en déduire la position du récepteur dans un référentiel géographique connu (WGS 84²).

La précision peut être améliorée par méthode différentielle (DGPS³), en s'aidant d'une station de référence proche de l'endroit où l'on effectue les mesures [14].

II.3.2 Géolocalisation par GPRS/GSM

Le GSM ou « *Global System for Mobile communications* » est une norme de transmission téléphonique et numérique utilisée par les téléphones sans fil.

Le positionnement par GSM permet d'obtenir une précision allant de 200 mètres à plusieurs kilomètres, selon si le terminal se trouve en milieu urbain ou rural.

La technique GSM Cell ID permet, à partir des positions géographiques des antennes auxquelles l'identifiant de cellule est connecté, de déterminer sa position et d'émettre une estimation quant à sa localisation.

II.3.3 Géolocalisation par Wi-Fi

La géolocalisation rapprochée via Wi-Fi, réalisée par triangulation, permet de situer des objets ou des personnes, en extérieur comme en intérieur. Sa précision est de l'ordre du mètre, mais dépend du délai et de la puissance des signaux entre l'objet et les bornes Wi-Fi. Toutefois, ces produits ne fournissent que des coordonnées brutes (x, y, z) qui doivent ensuite être interprétées par un moteur contextuel, proposé par des éditeurs. Le déploiement de ces

² **World Geodetic System 1984** : système géodésique standard mondial, utilisé notamment par le système GPS

³ C'est une amélioration du GPS. Il utilise un réseau de stations fixes de référence qui transmet l'écart entre les positions indiquées par les satellites et leurs positions réelles connues

outils est possible en entrepôt, mais des difficultés techniques peuvent apparaître en fonction des contraintes de l'environnement par exemple l'interférence avec les infrastructures métalliques.

II.3.4 Géolocalisation par adresse IP

Cette méthode permet de déterminer la position géographique d'un ordinateur ou de n'importe quel terminal connecté à Internet en se basant sur son adresse IP. Les adresses IP sont gérées par l'IANA⁴, une organisation qui s'occupe de découper les blocs d'adresses IP disponibles et de les distribuer de façon très contrôlée aux pays qui en demandent. Toutes ces attributions étant très bien documentées, il est possible de savoir dans quel pays se trouve un terminal connecté à Internet grâce à son adresse IP. On peut même obtenir un niveau de précision de l'ordre de la ville en se basant sur la distribution des adresses IP faite par les fournisseurs d'accès à Internet. Le Royaume-Uni constitue une exception notable car la précision de l'information accessible librement n'est pas la ville mais seulement la nation, ainsi une adresse IP du Sud-Est de l'Angleterre peut apparaître comme enregistrée au Nord-Ouest de l'Angleterre. Pour une adresse IP associée à une date et une heure données, les forces de police peuvent obtenir auprès du fournisseur d'accès à internet la position géographique exacte, le nom et l'adresse communiquées par le titulaire. Mais ces données ne sont conservées qu'un temps limité, dont la durée varie selon les pays. De même pour un espace commercial d'accès à Internet ("cybercafé"), selon les pays le commerçant peut être tenu de garder les dates, heures et documents d'identité scannés pendant une certaine période. [15]

II.3.5 Géolocalisation par RFID

La RFID (*Radio Frequency IDentification*) a pour objectif l'identification locale d'un objet grâce à une étiquette passive (qui n'émet pas d'informations par elle-même et réagit uniquement à proximité d'un lecteur) ou active (qui émet un signal d'elle-même et permet l'identification sur des distances plus importantes). Suite à l'émission d'un signal de radio fréquence, l'étiquette RFID passive (TAG) est alors en mesure de fournir ou d'enregistrer de l'information sur l'objet. Cela permet ainsi de déterminer quel objet ou produit traverse un point de passage, où se trouvent les capteurs RFID, mais aussi d'indiquer dans l'étiquette que

⁴ IANA est responsable de la coordination à l'échelle mondiale des adresses uniques d'Internet, et est exploité par la *Société Internet pour Assigned Names and Numbers* (ICANN). Voir aussi www.iana.org

l'objet est passé par tel point de passage. Les retours sur investissement peuvent être courts mais dépendent du niveau d'intégration de cette technologie tout au long de la chaîne logistique et de paramètres techniques tel que le taux de lecture effectif.

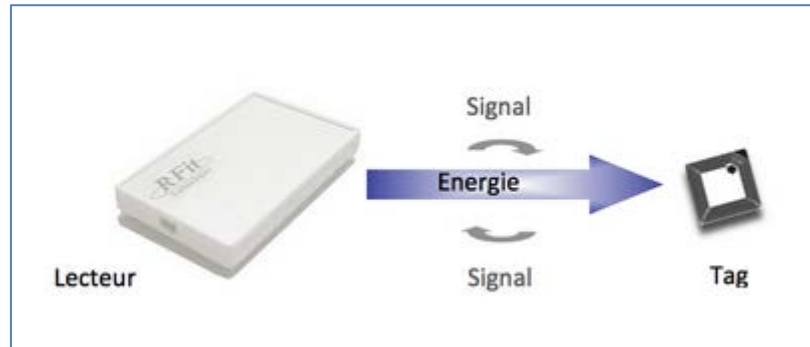


Figure II.2 Système RFID [16]

II.4 Combinaison des différentes techniques

Il y a plusieurs inconvénients à l'utilisation d'une seule technique de géolocalisation, nous citons :

- La dépendance au réseau GPS : l'incapacité de l'utiliser à l'intérieur (indoor) et le temps de réponse à l'allumage.
- La dépendance au réseau GSM : sa couverture géographique, l'accès au réseau GPRS pour exploiter l'information.
- La dépendance à la présence de bornes Wi-Fi

Des dispositifs qui combinent ces trois techniques peuvent être employés pour localiser un terminal dans n'importe quelle situation, existent. La précision de ce positionnement va varier en fonction des technologies disponibles, mais le temps de réponse à l'allumage et l'adaptabilité s'en verront améliorées. Ceci permet par exemple de géolocaliser une personne à l'extérieur en utilisant le GPS, et de garder sa trace à l'intérieur des bâtiments ou des tunnels en utilisant la technologie GSM couplée au WiFi pour plus de précision.

L'iPhone d'Apple est un exemple de terminal capable d'utiliser une méthode hybride de géolocalisation grâce à ses interfaces GSM/UMTS, WiFi et à son récepteur GPS. Cette fonction est fournie par la société *skyhookwireless*, qui en l'occurrence met à disposition les bases

de données appropriées pour transformer les identifiants des cellules GSM et des points d'accès WiFi en latitude/longitude et rayon de précision.

Cette couverture fonctionne très bien à l'extérieur (*outdoor*), en revanche ces systèmes sont inefficaces à l'intérieur (*indoor*). [17]

II.5 Plateformes logicielles de géolocalisation

II.5.1 Composants

Les composants essentiels d'une plateforme de géolocalisation sont les suivants :

- **Terminal communicant** : C'est le terminal qui reçoit ses coordonnées géographiques (via GPS ou tout autre moyen) et qui les envoie via un réseau de télécommunications à la plateforme;
- **Système informatique capable de recevoir, stocker et traiter les informations** : il s'agit des serveurs informatiques qui hébergent l'infrastructure et qui reçoivent et traitent les données envoyées par les terminaux. Ce sont les mêmes serveurs qui mettront à disposition l'information aux utilisateurs (via une interface web par exemple);
- **Module cartographique** : c'est le module intégré au système informatique qui va permettre d'afficher la position des terminaux sur un fond cartographique adapté. Ce module prend en charge les calculs de distances, d'itinéraires, détecte l'interaction avec les zones et permet d'avoir accès à des informations terrain (sens interdits, restrictions pour les poids lourds, vitesses autorisées...).

II.5.2 Architecture temps réel

La position géographique d'un terminal géolocalisé reste néanmoins une information-brute qui peut être exploitée et couplée à d'autres données afin de créer une vaste quantité de services à forte valeur ajoutée.

Afin d'exploiter cette information, la donnée (position) générée par un terminal qui se trouve sur le terrain doit être transmise à une plateforme logicielle qui va la traiter, la présenter graphiquement à l'utilisateur et l'associer à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou de la flotte de terminaux.

Voici les étapes de la chaîne de traitement :

Le terminal détermine sa position géographique grâce à une des techniques de géolocalisation citées précédemment (de préférence GPS, GSM et/ou WiFi);

Il envoie ces données vers une plateforme logicielle soit par le réseau GSM/GPRS soit par un réseau satellitaire de type Inmarsat⁵;

La plateforme logicielle de géolocalisation traite la donnée et positionne le terminal géographiquement sur une carte moyennant la précision offerte par la technique de géolocalisation utilisée. De plus, en combinant plusieurs informations, notamment récupérées via un système de télérelève (trafic routier, autonomie du véhicule, points à visiter etc...), des calculs d'itinéraires ou de tournées peuvent par exemple être générés.

Cette carte ainsi que tous les traitements effectués sont mis à disposition de l'utilisateur à travers un portail web hébergé sur un serveur accessible depuis Internet, ou à travers une application métier installée sur le poste de travail.

Pour transmettre les différentes informations récupérées par le terminal (position géographique) nous recensons deux principaux moyens de transmission : le réseau GSM/GPRS et le réseau satellitaire.

II.6 Types de terminaux existants

II.6.1 Modes de fonctionnement

Typiquement, les terminaux de géolocalisation existants peuvent être classés dans une de ces 3 catégories, même si certains peuvent être configurés pour fonctionner dans un mode au choix :

- **Data loggers** : Ces terminaux stockent les positions localement et celles-ci doivent ensuite être extraites ;
- **Data pullers** : Ces terminaux envoient leur position à la demande;
- **Data pushers** : Ces terminaux effectuent des envois fréquents de leur position.

⁵ Voir www.inmarsat.com

a) Data loggers

Ces terminaux sont généralement équipés de récepteurs GPS et se limitent à stocker dans leur mémoire interne leur position à des intervalles réguliers. Certains loggers GPS disposent de slots pour carte mémoire et/ou d'une mémoire interne ainsi qu'un port USB (le terminal est vu comme une clé USB). Ceci permet, postérieurement, de télécharger les données sur un ordinateur pour les traiter.

Ce genre de terminaux est surtout utilisé par des sportifs (joggers, amateurs de VTT...) qui téléchargent ensuite les données sur leur ordinateur personnel afin de calculer la durée du parcours ou d'afficher les points sur une carte à l'aide d'un logiciel SIG. Pour des parcours longs et non contrôlés, ce genre de dispositif peut permettre de déterminer le gagnant d'une épreuve et de déterminer si celui-ci est bien passé par des points de contrôle.

Ces terminaux peuvent aussi être combinés à des appareils numériques afin de géotagger les photos en se basant sur l'heure de la prise. Ces terminaux peuvent aussi être utilisés dans le cadre de surveillances ou de suivi de véhicules où l'émission de données serait impossible ou repérable.

b) Data pullers

Contrairement aux dispositifs de type "push", les data pullers se limitent à envoyer l'information uniquement sur demande. Ces dispositifs sont suffisants dans les cas où la position de l'objet ou la personne n'a pas besoin d'être connue en continu. Par exemple, la position du véhicule ne sera nécessaire que si celui-ci est volé.

De plus, cette méthode permet aux opérateurs de télécommunications de commercialiser un service de géolocalisation à travers des terminaux mobiles dépourvus de récepteurs GPS et de forfaits data. Il suffit d'avoir le numéro de téléphone du mobile et d'y être autorisé pour envoyer une requête de demande de position. L'opérateur localise alors la position du terminal grâce à la technique du Cell ID et vous envoie la position de celui-ci.

c) Data pushers

Ce sont les terminaux les plus utilisés pour des applications professionnelles. Ces terminaux envoient leur position à intervalles réguliers et programmables vers une plateforme de géolocalisation qui traite la donnée en temps réel.

Parmi les terminaux capables d'être géolocalisés et de transmettre l'information en temps réel, nous distinguons :

- Les terminaux mobiles de type PDA ou smartphones équipés d'un GPS et/ou d'un modem GSM/GPRS ;
- Les boîtiers de géolocalisation dédiés embarquant un récepteur GPS, un dispositif de télécommunications (GSM/GPRS ou satellite) et avec une capacité optionnelle de télérelève ;
- Les dispositifs de poche destinés aux personnes et embarquant un récepteur GPS et un modem GSM/GPRS ou satellite.

Ces dispositifs peuvent nécessiter le branchement à une source électrique ou être autonomes grâce à une batterie interne. En fonction de l'utilisation du terminal, celui-ci pourra être connecté à la batterie d'un véhicule ou disposer d'une grande autonomie (ex : suivi d'objets sans sources électriques comme des conteneurs, colis, animaux, remorques...).

II.6.2 Modes de communication

Voici les différents types de terminaux disponibles selon leur mode de transmission des données.

a. Terminaux GSM/GPRS

Ce moyen de transmission nécessite un terminal disposant d'un modem GSM/GPRS ou 3G/UMTS, ainsi que d'une carte SIM de n'importe quel opérateur avec un forfait "données" (Data) adapté. Le terminal nécessite d'être sous couverture GSM/GPRS pour pouvoir envoyer les données vers la plateforme de traitement. Ce type de terminal est utilisé lorsque l'objet ou la personne à géolocaliser reste dans une zone bien couverte par les réseaux GSM/GPRS.

b. Terminaux satellite

Ce type de terminaux envoie les données à travers un réseau de satellites de télécommunications comme Inmarsat. Même si ces types de canaux sont plus restrictifs au niveau de la quantité de données envoyées, ils peuvent offrir par ailleurs une couverture mondiale sans frais supplémentaires en fonction des satellites et protocoles utilisés. Cela en fait des termi-

naux idéaux pour la géolocalisation de conteneurs, navires ou véhicules circulant en plein désert.

Lorsque le terminal satellite est autonome grâce à une batterie interne, il peut fonctionner jusqu'à 7 ans sans nécessité d'en remplacer la batterie. Son autonomie varie alors en fonction de la fréquence d'actualisation souhaitée.

Pour des parcours de longues distances ou des remontées d'information peu fréquentes, les solutions satellites sont plus économiques que les solutions GSM/GPRS.

II.7 Conclusion

Dans ce chapitre nous avons décrit les différentes techniques de géolocalisation. Notre choix s'est porté la localisation par GPS qui est au cœur de notre solution.

Chapitre III :

Analyse et conception

III.1 Introduction

Les endroits les plus fréquemment sollicités dans les hôpitaux sont les services des urgences médicales. Ces lieux nécessitent la mobilisation continue des personnels médicaux. Leurs gestions est de plus en plus âpre du fait de la croissance démographique grandissante en nette progression et l'augmentation de la population du troisième âge notamment.

III.2 Problématique

L'urgence est un combat contre le temps. En effet, observer l'affolement et la détresse des proches de malades et les va-et-vient incessants des infirmiers et médecins d'un service à l'autre occasionnent beaucoup de désordre à l'intérieur du service des urgences et rendent peu efficace la prise en charge des patients.

Une des contraintes de l'efficacité d'intervention, c'est de trouver le bon médecin, au bon endroit et au bon moment lors d'une urgence donnée surtout si ce dernier ne se trouve pas nécessairement à l'intérieur du service auquel il est affecté. Notre objectif consiste à automatiser cette recherche en localisant les médecins à travers leurs Smartphones et les triant par service. Ceci aidera les réceptionnistes à mieux gérer les urgences, atténuer leurs charges de travail, gagner du temps et par conséquent améliorer la prise en charge des malades.

Dans cette phase, nous décrirons la démarche suivie pour mener à bien notre travail. Il s'agit de mettre en place une application de gestion d'alertes destinées aux réceptionnistes et aux médecins, dépendants d'un hôpital, qui ne se trouvent pas nécessairement sur leurs lieux de travail, afin de répondre instantanément et efficacement aux urgences signalées.

L'application qui sera mise en place doit pouvoir sélectionner les médecins joignables triés par leurs services et leurs spécialité. L'application localisera les médecins via leurs Smartphones dotés du système de localisation GPS.

Le système ne doit être, en aucun cas, un mouchard qui suit le mouvement des médecins mais seulement pour répondre à un impératif d'efficacité dans la prise en charge des malades.

III.3 Rappels sur l'UML

III.3.1 Définition

UML (*Unified Modeling Language*) standardisé par l'OMG¹) est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch², OMT³, OOSE. [18]

UML sert à :

- Décomposer le processus de développement;
- Mettre en relation les experts métiers et les analystes;
- Coordonner les équipes d'analyse et de conception;
- Séparer l'analyse de la réalisation;
- Prendre en compte l'évolution de l'analyse et du développement;
- Migrer facilement vers une architecture objet d'un point de vue statique et dynamique.

III.3.2 Le formalisme d'UML

UML se décompose en plusieurs sous-ensembles

- **Les vues** : Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir ou retrouver le système complet.

- **Les diagrammes** : Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.

¹ *Object Management Group*, association américaine à but non lucratif créée pour la standardisation et la promotion du modèle objet sous toutes ses formes.

² Grady Booch est le créateur d'une approche d'analyse et de conception orientée objet portant son nom.

³ OMT (*Object Modeling Technique*) est une technique de modélisation objet.

- **Les modèles d'élément** : Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagramme. Exemple d'élément : cas d'utilisation, classe, association, etc.

III.3.3 Définition architecture avec UML

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système.

- **La vue logique** : cette vue concerne « l'intégrité de conception ».

Cette vue de haut niveau se concentre sur l'abstraction et l'encapsulation, elle modélise les éléments et mécanismes principaux du système.

Elle identifie les éléments du domaine, ainsi que les relations et interactions entre ces éléments :

- les éléments du domaine sont liés au(x) métier(s) de l'entreprise,
- ils sont indispensables à la mission du système,
- ils gagnent à être réutilisés (ils représentent un savoir-faire).

Cette vue organise aussi (selon des critères purement logiques), les éléments du domaine en "catégories":

- pour répartir les tâches dans les équipes,
- regrouper ce qui peut être générique,
- isoler ce qui est propre à une version donnée, etc...

- **La vue des composants** : Cette vue concerne « l'intégrité de gestion ».

Cette vue de bas niveau (aussi appelée "vue de réalisation"), montre : L'allocation des éléments de modélisation dans des modules (fichiers sources, bibliothèques dynamiques, bases de données, exécutables, etc...). En d'autres termes, cette vue identifie les modules qui réalisent (physiquement) les classes de la vue logique. L'organisation des composants, c'est-à-dire la distribution du code en gestion de configuration, les dépendances entre les composants... Les contraintes de développement (bibliothèques externes...). La vue des composants montre aussi l'organisation des modules en "sous-systèmes", les interfaces des sous-systèmes et leurs dépendances (avec d'autres sous-systèmes ou modules).

- **La vue des processus** : Cette vue concerne « l'intégrité d'exécution ».
Cette vue est très importante dans les environnements multitâches ; elle montre :
 - La décomposition du système en termes de processus (tâches).
 - Les interactions entre les processus (leur communication).
 - La synchronisation et la communication des activités parallèles (threads).
- **La vue de déploiement** : Cette vue concerne « l'intégrité de performance ».

Cette vue très importante dans les environnements distribués, décrit les ressources matérielles et la répartition du logiciel dans ces ressources :

- La disposition et nature physique des matériels, ainsi que leurs performances.
 - L'implantation des modules principaux sur les nœuds du réseau.
 - Les exigences en termes de performances (temps de réponse, tolérance aux fautes et pannes...).
- **La vue des besoins des utilisateurs** : Cette vue (dont le nom exact est "vue des cas d'utilisation"), guide toutes les autres. Dessiner le plan (l'architecture) d'un système informatique n'est pas suffisant, il faut le justifier ! Cette vue définit les besoins des clients du système et centre la définition de l'architecture du système sur la satisfaction (la réalisation) de ces besoins. A l'aide de scénarios² et de cas d'utilisation, cette vue conduit à la définition d'un modèle d'architecture pertinent et cohérent. Cette vue est la "colle" qui unifie les quatre autres vues de l'architecture. Elle motive les choix, permet d'identifier les interfaces critiques et force à se concentrer sur les problèmes importants.

La relation entre les différentes perspectives a été représentée par Ph. Kruchten⁴ dans le schéma suivant, dit « schéma 4+1 vues ».

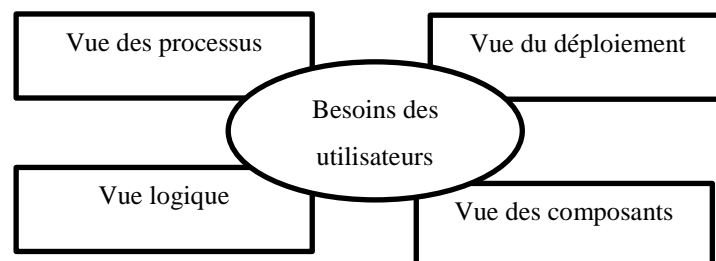


Figure III.1. Schéma de principe de l'encapsulation

⁴ Professeur de génie logiciel à l'Université de la Colombie- Britannique, USA.

III.3.4 Les Diagrammes d'UML

- Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. C'est une perspective du modèle, pas "le modèle".
- Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis).
- Un type de diagramme UML véhicule une sémantique précise (un type de diagramme offre toujours la même vue d'un système).
- Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système.
- Par extension et abus de langage, un diagramme UML est aussi un modèle (un diagramme modélise un aspect du modèle global).

III.4 Analyse

Dans cette étape, nous allons recenser les besoins et des réceptionnistes exerçant aux urgences médicale et formaliser ainsi l'ensemble des contraintes.

III.4.1 Spécification des besoins

Notre travail porte sur la conception et réalisation d'une application d'assistance aux urgences médicales dans un hôpital. Il consiste notamment en la gestion des alertes destinées aux médecins. Elle doit offrir un environnement interactif aux différents acteurs qui seront décrits dans la section suivante. Elle permet de gérer l'ensemble des médecins spécialistes disponibles et leurs envoyer des alertes en fonction des différents cas d'urgence.

III.4.2. Architecture générale du système

Nous présentons dans ce qui suit une ébauche du système qui sera mis en place. Afin d'avoir une vue globale du fonctionnement de celui-ci, nous l'avons décomposé en 3 modules :

- Module d'authentification.
- Module de travail.
- Module de résultats.

La figure III.2 montre l'architecture générale du système :

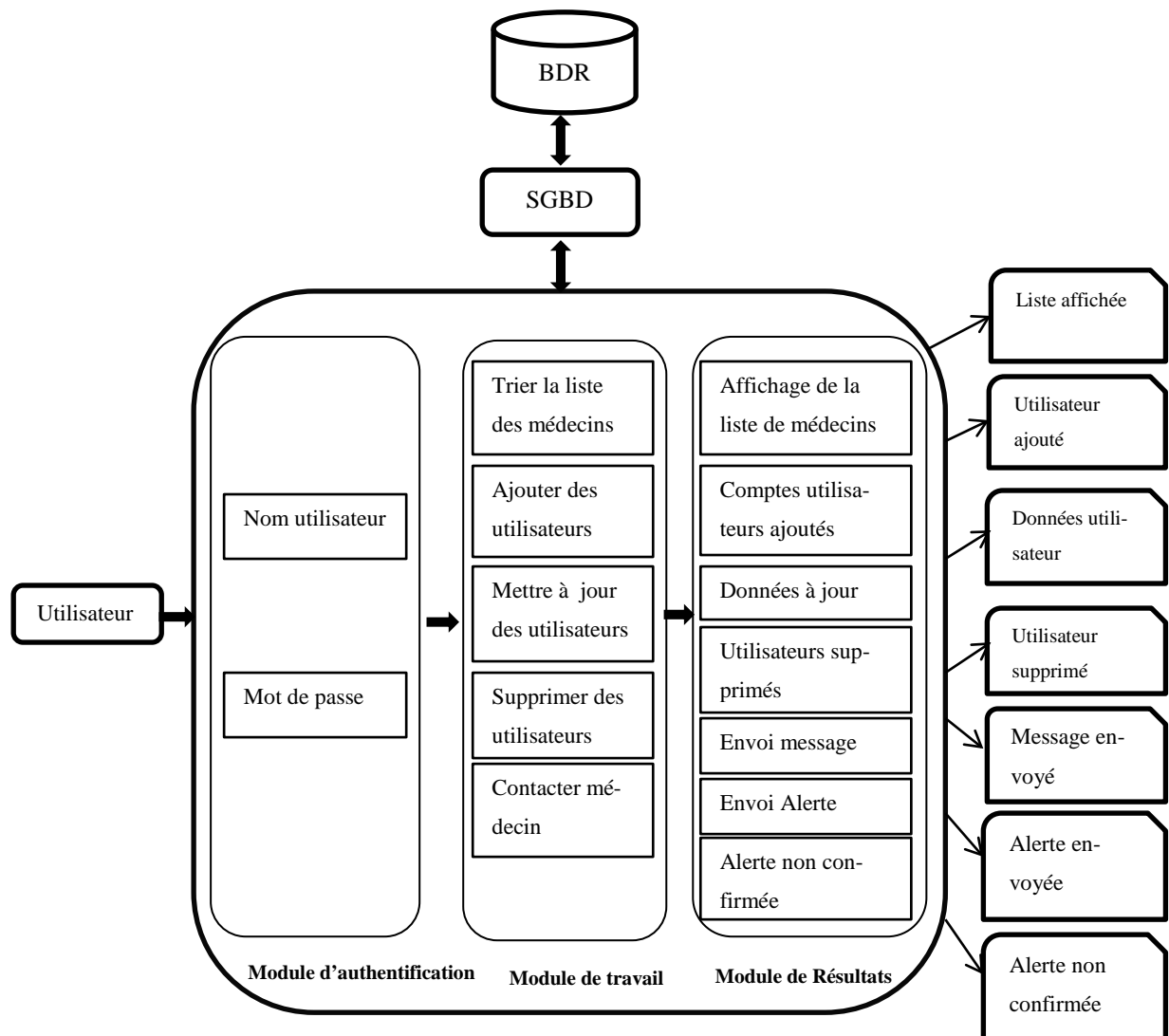


Figure III.2. Architecture générale du système

1. Module d'authentification :

Ce module permet à l'utilisateur de s'authentifier en fournissant son nom d'utilisateur et son mot de passe et ainsi pouvoir accéder à son espace autorisé par le système.

2. Module de travail :

Ce module regroupe les différentes fonctionnalités que propose notre système énumérées dans ce qui suit :

- **Trier la liste des médecins :** le système affiche la liste de médecins joignables triés par service.
- **Ajouter des utilisateurs:** il y a trois catégories d'utilisateurs, l'administrateur, les médecins et les réceptionnistes.
- **Mettre à jour des utilisateurs :** pour modifier une ou plusieurs informations concernant un utilisateur donné.
- **Supprimer des utilisateurs :** permet de supprimer un utilisateur.
- **Contacter des médecins :** permet de contacter un médecin en le sélectionnant depuis la liste triée par service et lui envoyer une alerte ou un message.

3. Module de résultats :

Il représente les résultats du module de travail. Il comprend :

- l'affichage de la liste de médecins à contacter
- confirmation d'ajout ou de suppression d'un utilisateur
- Confirmation de mise à jour des données d'un utilisateur
- Confirmation d'envoi d'un message ou d'une alerte

- Génération d'un message « Alerte non confirmée par le médecin ».dans le cas où le médecin contacté n'a pas envoyé son « OK » suite à une alerte.

III.4.3 Identification des acteurs

Un acteur représente une entité qui interagit avec le système. Les utilisateurs de notre application sont :

- Les réceptionnistes
- Les médecins
- L'administrateur de l'application

III.4.4. Les cas d'utilisation

Acteur	Cas d'utilisation
Réceptionniste	<ul style="list-style-type: none">- S'authentifier- Visualiser la liste des médecins triée par service- Sélectionner un médecin- Envoyer une alerte au médecin sélectionné- Envoyer un message au médecin sélectionné- Se déconnecter
Médecin	<ul style="list-style-type: none">- S'authentifier- Répondre à une alerte- Répondre à un message- Se déconnecter

Administrateur	<ul style="list-style-type: none"> - S'authentifier - Ajouter/modifier/supprimer un service - Ajouter/modifier/supprimer un médecin - Ajouter/modifier/supprimer un réceptionniste - Se déconnecter
----------------	--

Tableau III.1 Les cas d'utilisation

III.4.5. Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

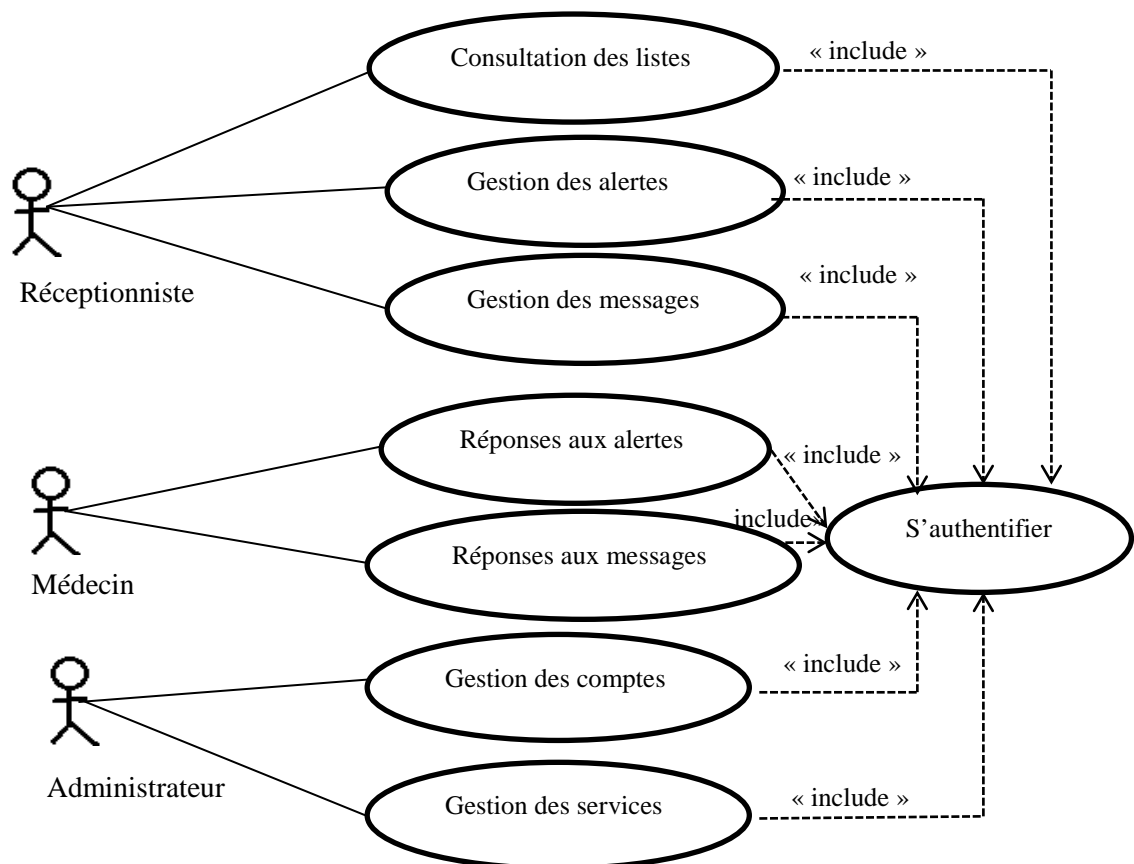


Figure III.3 Diagramme de cas d'utilisation général

Pour mieux voir les différentes fonctionnalités de notre application, nous avons élaboré des diagrammes de cas d'utilisation détaillés pour les trois types d'utilisateurs de notre application.

A- Diagramme de cas d'utilisation du réceptionniste:

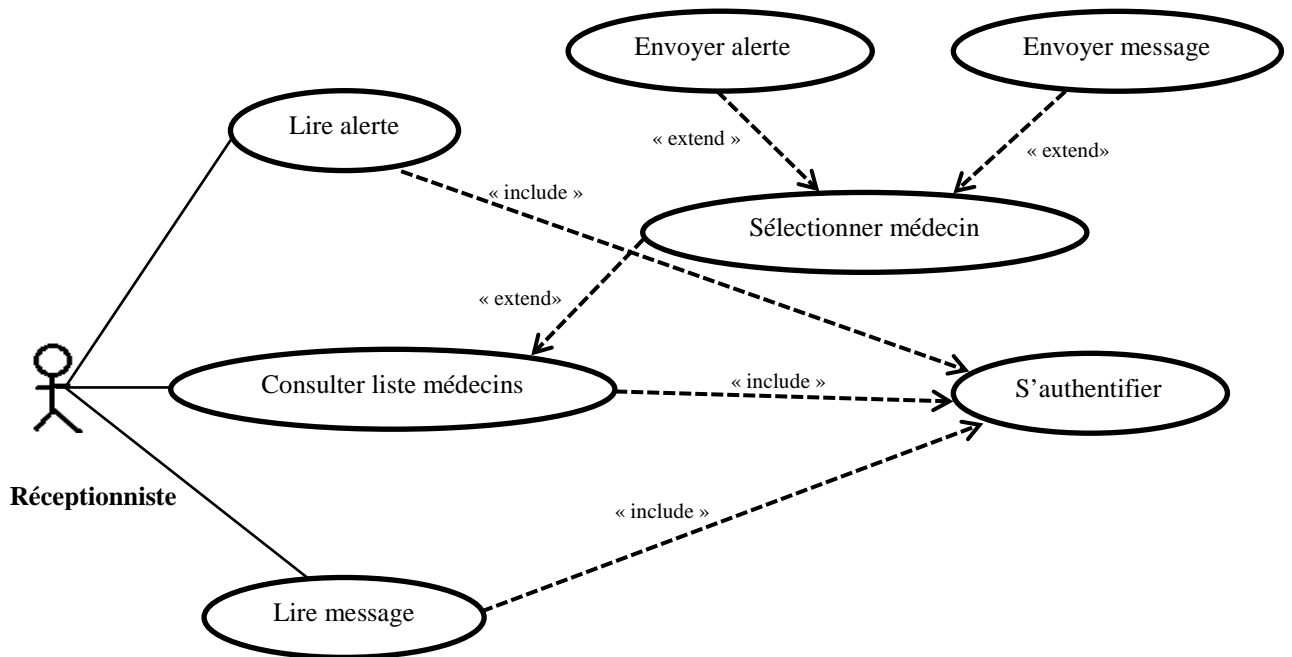


Figure III.4. Diagramme de cas d'utilisation du réceptionniste

B- Diagramme de cas d'utilisation du médecin :

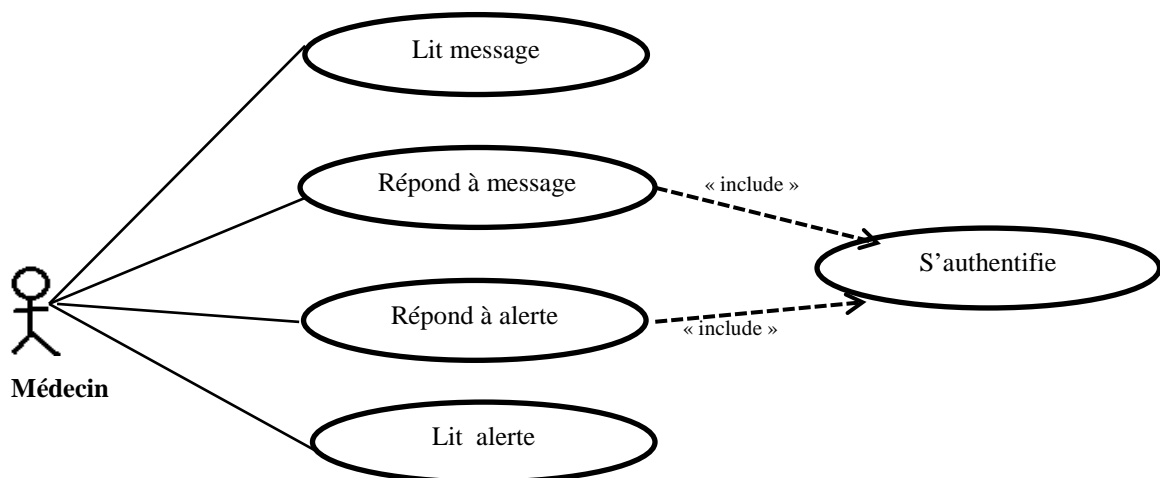


Figure III.5 Diagramme de cas d'utilisation du médecin.

C- Diagramme de cas d'utilisation de l'administrateur :

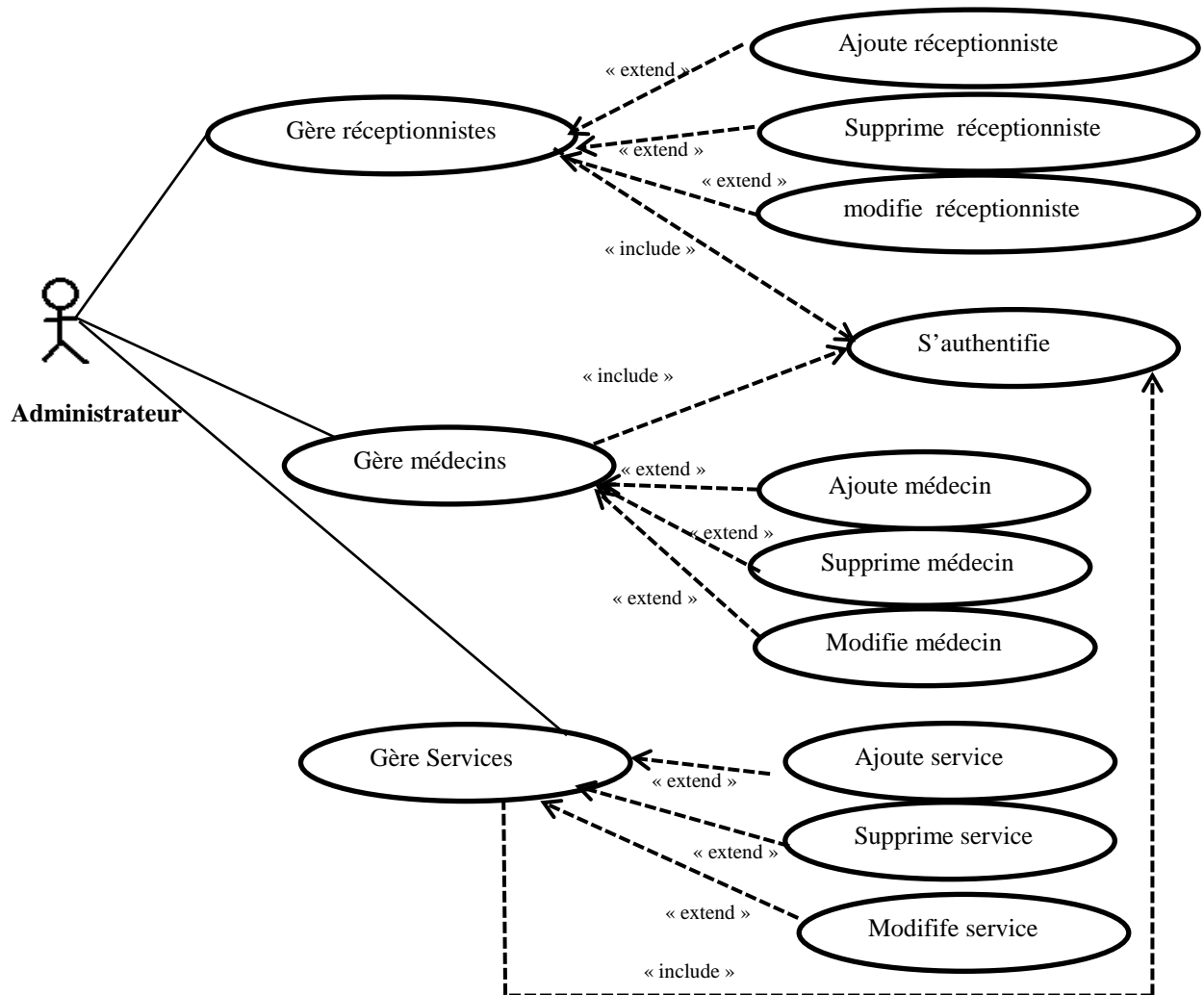


Figure III.6 Diagramme de cas d'utilisation de l'administrateur.

III.4.6 Identification des scénarios

Un scénario représente une succession particulière d'enchaînement, qui s'exécute du début jusqu'à la fin du cas d'utilisation [19]. Un cas d'utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs (qui se terminent d'une façon normale) ou d'erreurs (qui se terminent en échec).

Dans ce qui suit, nous présenterons dans des tableaux, les différents scénarios associés à chaque acteur.

III.4.6.1 Les scénarios par tâches d'un réceptionniste

Le tableau III.2 montre les différentes tâches et les scénarios correspondants d'un réceptionniste.

Tâche	Scénarios
T01 : S'authentifier	<p>S01 : L'utilisateur demande la connexion.</p> <p>S02 : Le système lui affiche un formulaire d'authentification.</p> <p>S03 : L'utilisateur saisit son nom utilisateur et son mot de passe, puis il clique sur « Se connecter ».</p> <p>S04 : Le système vérifie la validité des informations reçues et affiche la page d'accueil de l'utilisateur, sinon il retourne un message d'erreur</p>
T02 : Envoyer une alerte	<p>S05 : L'utilisateur clique sur un service</p> <p>S06 : L'utilisateur clique sur le médecin, le plus proche, à contacter</p> <p>S07 : L'utilisateur clique sur « Alerte »</p>
T03 : Envoyer un message	<p>S08 : L'utilisateur clique sur un service</p> <p>S09 : L'utilisateur clique sur un médecin à contacter</p> <p>S10 : L'utilisateur clique sur « Message » s'il s'agit d'un simple message d'information</p> <p>S11 : L'utilisateur écrit le message</p> <p>S12 : L'utilisateur clique sur « Envoyer » pour envoyer le message</p>

Tableau III.2 Récapitulatif des scénarios par tâches d'un réceptionniste.

III.4.6.2 Les scénarios par tâches d'un médecin

Le tableau III.3 montre les différentes tâches et les scénarios correspondants d'un médecin

Tâche	Scénarios
T04 : S'authentifier	S01 : L'utilisateur demande la connexion. S02 : Le système lui affiche un formulaire d'authentification. S03 : L'utilisateur saisit son nom utilisateur et son mot de passe, puis il clique sur « Se connecter ». S04 : Le système vérifie la validité des informations reçues et affiche la page d'accueil de l'utilisateur, sinon il retourne un message d'erreur
T05 : Répondre favorablement à une alerte	S13 : L'utilisateur lit le contenu de l'alerte S14 : L'utilisateur confirme sa venue en cliquant sur « J'arrive »
T06 : Répondre défavorablement à une alerte	S15 : L'utilisateur lit le contenu de l'alerte S16 : L'utilisateur répond défavorablement à l'alerte indiquant son indisponibilité en cliquant sur « Occupé »
T07 : Ne pas répondre à une alerte	S17 : L'utilisateur ne répond pas au bout d'un « time out », le système affiche un message pour le réceptionniste « Le médecin contacté ne répond pas, veuillez contacter un autre »
T08 : Répondre à un message	S18 : L'utilisateur clique sur « Répondre » S19 : L'utilisateur tape le message S20 : L'utilisateur envoie le message

Tableau III.3 Récapitulatif des scénarios par tâches d'un médecin.

III.4.6.3 Les scénarios par tâches d'un administrateur

Le tableau III.4 montre les différentes tâches et les scénarios correspondants d'un administrateur.

Tâches	Scénarios
T09 : S'authentifier	<p>S01 : L'utilisateur demande la connexion.</p> <p>S02 : Le système lui affiche un formulaire d'authentification.</p> <p>S03 : L'utilisateur saisit son nom utilisateur et son mot de passe, puis il clique sur « Se connecter ».</p> <p>S04 : Le système vérifie la validité des informations reçues et affiche la page d'accueil de l'utilisateur, sinon il retourne un message d'erreur</p>
T10 : Ajouter un nouveau réceptionniste	<p>S21 : L'utilisateur demande l'ajout d'un réceptionniste en cliquant sur « Ajouter un réceptionniste ».</p> <p>S22 : Le système affiche un formulaire d'ajout de réceptionniste.</p> <p>S23 : L'utilisateur remplit le formulaire et valide l'ajout en cliquant sur « Enregistrer ».</p> <p>S24 : Le système vérifie la validité des informations reçues et affiche un message pour confirmer l'ajout.</p>
T11 : Ajouter un nouveau médecin	<p>S24 : L'utilisateur demande l'ajout d'un médecin en cliquant sur « Ajouter un médecin ».</p> <p>S25 : Le système affiche un formulaire d'ajout de médecin.</p> <p>S26 : L'utilisateur remplit le formulaire et valide l'ajout en cliquant sur « Enregistrer ».</p> <p>S27 : Le système vérifie la validité des informations reçues et affiche un message pour confirmer l'ajout.</p>

<p>T12 : Supprimer un réceptionniste</p>	<p>S28 : L'utilisateur sélectionne un réceptionniste à supprimer</p> <p>S29 : L'utilisateur clique sur « Supprimer »</p> <p>S30 : L'utilisateur clique sur « OK » pour confirmer la suppression.</p> <p>S31 : L'utilisateur clique sur « Annuler » pour annuler la suppression.</p>
<p>T13 : Supprimer médecin.</p>	<p>S32 : L'utilisateur sélectionne un médecin à supprimer</p> <p>S33 : L'utilisateur clique sur « Supprimer »</p> <p>S34 : L'utilisateur clique sur « OK » pour confirmer la suppression.</p> <p>S35 : L'utilisateur clique sur « Annuler » pour annuler la suppression.</p>
<p>T14 : Mettre à jour un réceptionniste</p>	<p>S36 : L'utilisateur affiche la liste des réceptionnistes.</p> <p>S37 : L'utilisateur sélectionne un réceptionniste</p> <p>S38 : L'utilisateur clique sur « Modifier les informations »</p> <p>S39 : Le système lui affiche le formulaire du réceptionniste.</p> <p>S40 : L'utilisateur modifie les informations.</p> <p>S41 : L'utilisateur clique sur « Valider ».</p> <p>S42 : L'utilisateur clique sur « OK » pour confirmer les modifications.</p> <p>S43 : L'utilisateur clique sur « Annuler » pour annuler les modifications.</p>

T15 : Mettre à jour un médecin	S44 : L'utilisateur affiche la liste des médecins. S45 : L'utilisateur sélectionne un médecin S37 : L'utilisateur clique sur « Modifier les informations » S46 : Le système lui affiche le formulaire du médecin. S47 : L'utilisateur modifie les informations. S48 : L'utilisateur clique sur « Valider ». S49 : Le système lui affiche : Cliquer sur « Annuler » pour annuler les modifications ou sur « Confirmer » pour confirmer les modifications. S50 : L'utilisateur clique sur « OK » pour confirmer les modifications. S51 : L'utilisateur clique sur « Annuler » pour annuler les modifications.
---------------------------------------	---

Tableau III.4 Récapitulatif des scénarios par tâches de l'administrateur.

III.5. Conception

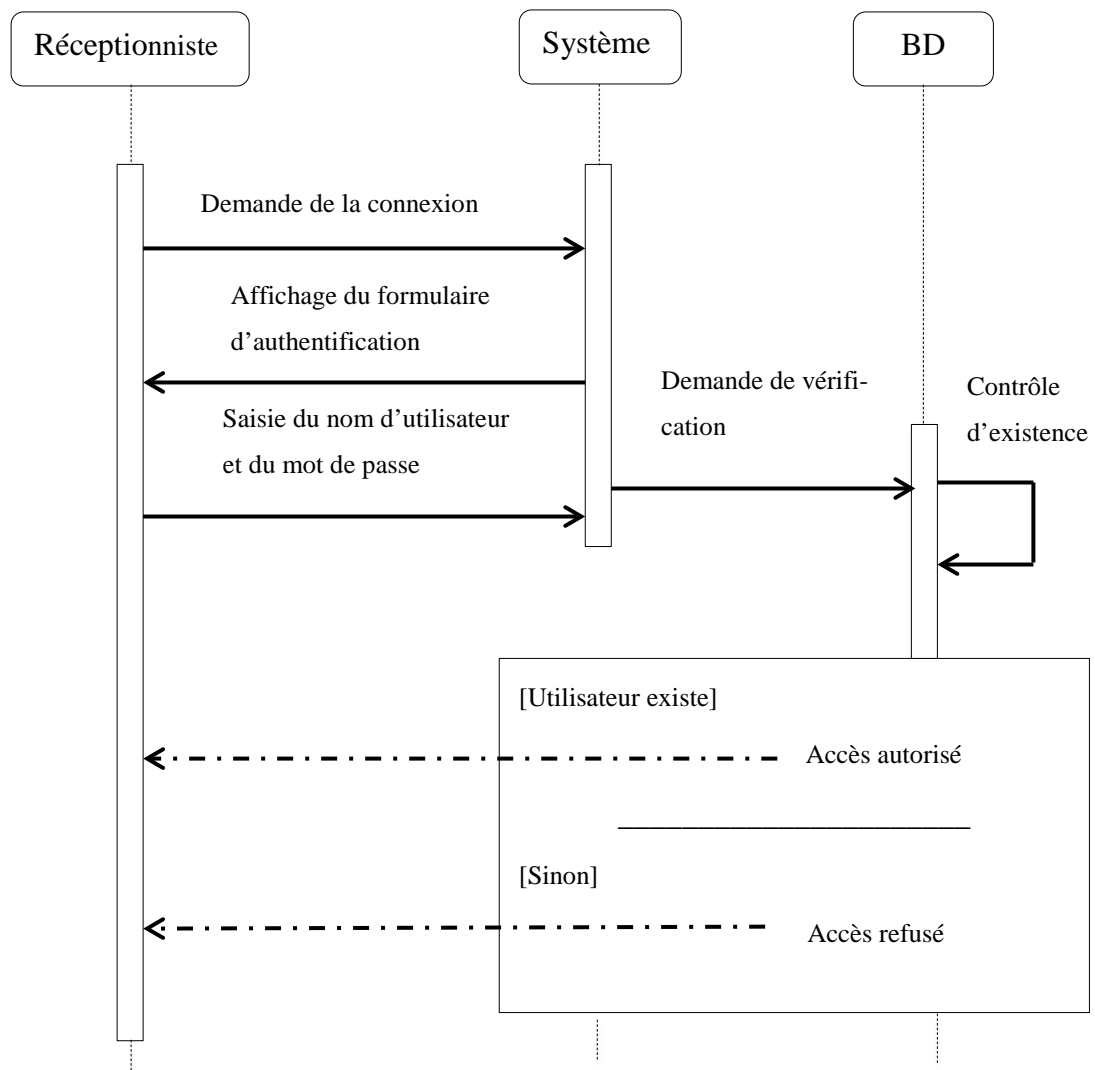
La conception est l'étape qui suit l'analyse. Elle consiste à modéliser et à détailler tous les éléments de modélisation issus de la phase d'analyse.

III.5.1 Diagramme de séquence

Le diagramme de séquence montre des interactions entre objets selon un point de vue temporel. Ce type de diagramme sert à modéliser les aspects dynamiques des systèmes et des scénarios complexes mettant en œuvre peu d'objets.

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique [20].

Via la description des cas d'utilisation nous pouvons identifier les scénarios. Dans ce qui suit nous allons traduire des exemples de scénarios en diagrammes de séquence.

1. Diagramme de séquence simple de cas d'utilisation « S'authentifier ».**Figure III.7.** Diagramme de séquence simple de cas d'utilisation « S'authentifier ».

2. Diagramme de séquence de cas d'utilisation « Ajouter un nouveau réceptionniste » :

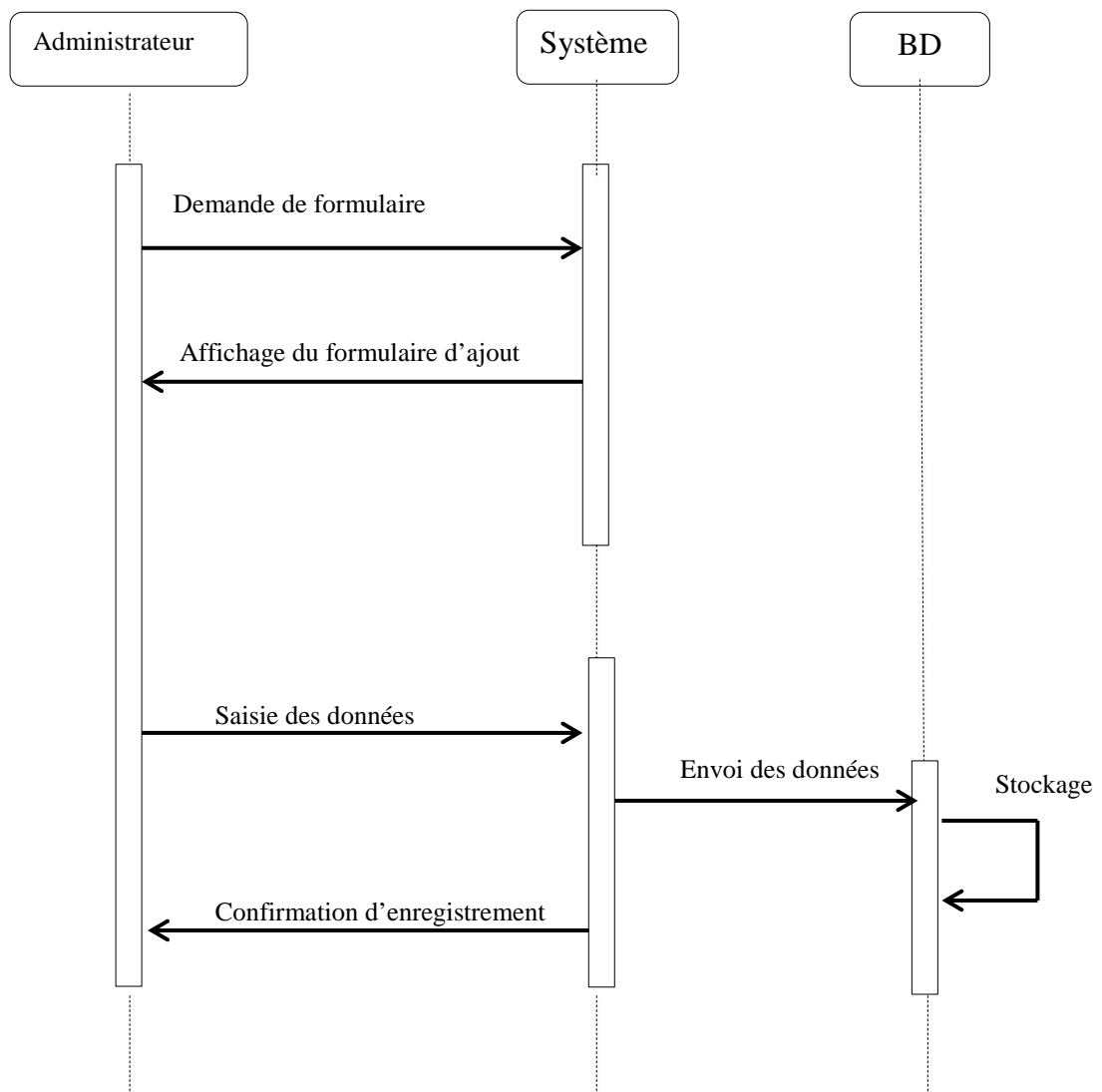


Figure III.8. Diagramme de séquence simple de cas d'utilisation « Ajouter un nouveau réceptionniste ».

3. Diagramme de séquence de cas d'utilisation «Supprimer médecin » :

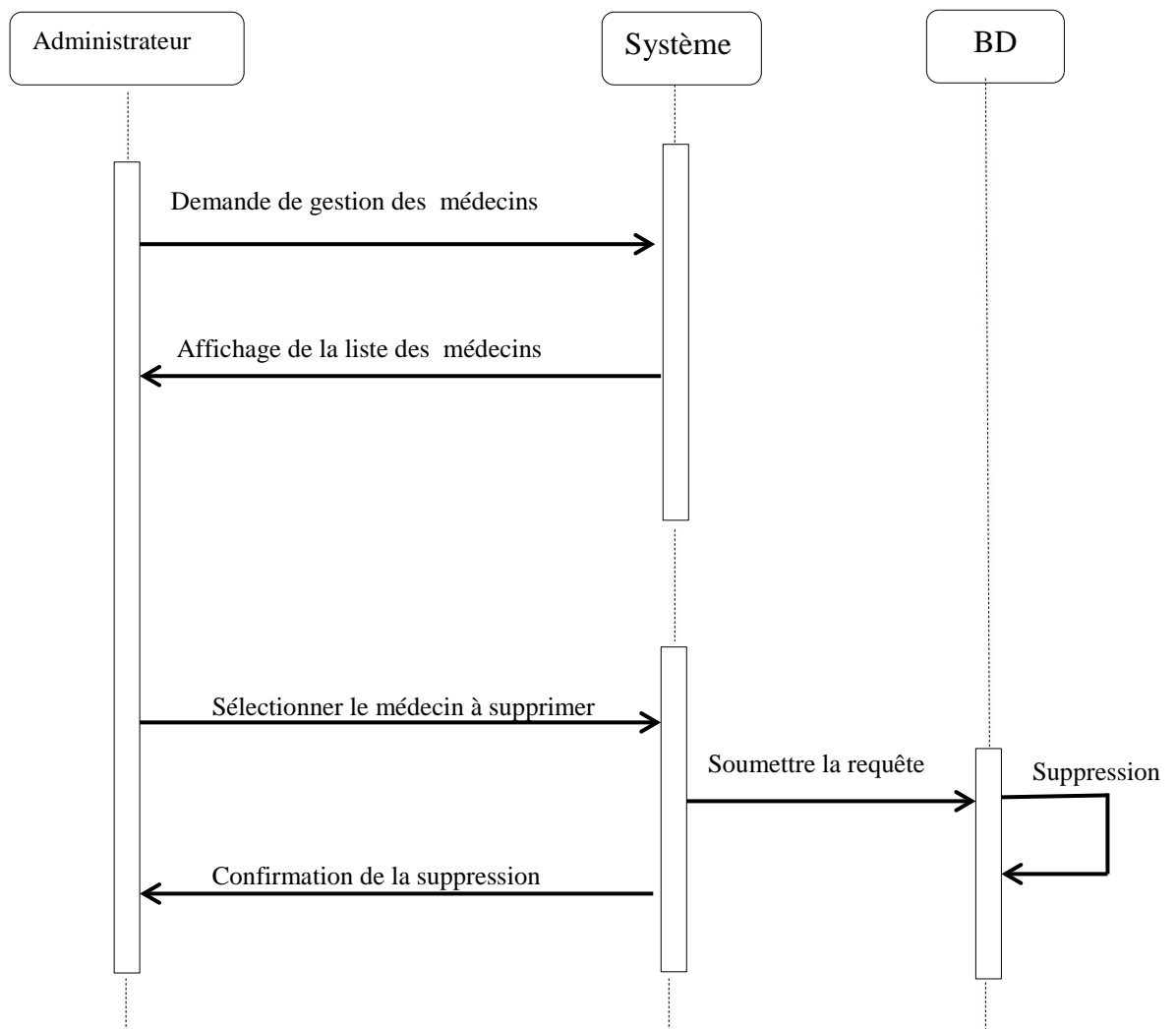


Figure III.9. Diagramme de séquence simple de cas d'utilisation « Supprimer un médecin ».

III.5.2 Diagramme des classes

Le diagramme de classe montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

III.5.2.1 Les règle de gestion

Les relations entre les tables doivent respecter les relations suivantes :

- Un médecin peut avoir un ou plusieurs messages et un message concerne un et un seul médecin ;
- Un médecin peut avoir un ou plusieurs alertes et une alerte concerne un et un seul médecin ;
- Un médecin est affecté à un et un seul service et à un service peuvent être affectés un ou plusieurs médecins ;
- Un message est envoyé à un et un seul médecin et un médecin peut recevoir un ou plusieurs messages ;
- Une alerte est envoyée à un et un seul médecin et un médecin peut recevoir une ou plusieurs alertes ;
- Un réceptionniste peut envoyer un ou plusieurs message et un message est envoyé par un et un seul réceptionniste ;
- Un réceptionniste peut envoyer une ou plusieurs alertes et une alerte est envoyée par un et un seul réceptionniste ;
- Un administrateur gère un ou plusieurs services et un service est géré par un et un seul administrateur ;
- Un administrateur gère un ou plusieurs médecins et un médecin est géré par un et un seul administrateur ;
- Un administrateur gère un ou plusieurs réceptionnistes et un réceptionniste est géré par un et un seul administrateur ;
- Un message peut être un « message simple » ou une alerte.

Remarque : Nous appelons message simple tout message d'information qui n'exprime pas une urgence médicale.

Le diagramme de classes global est représenté dans la figure III.10 :

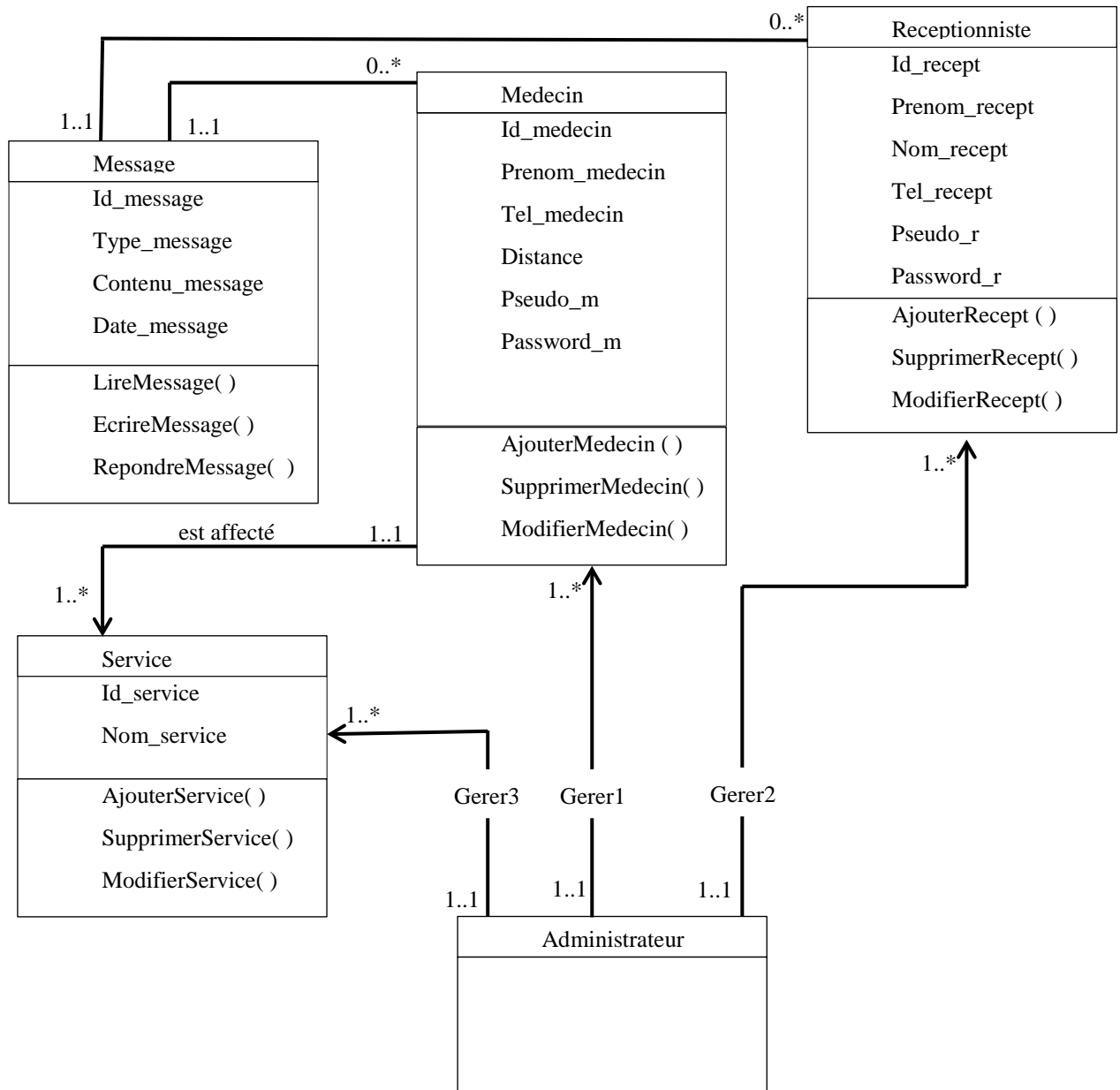


Figure III.10. Diagramme de classe global.

III.5.3 Conception de la base de données

Notre base de données est constituée des tables suivantes :

1. Table Medecin

Attribut	Description	Type	Clé
Id_medecin	Numéro d'identification du médecin	Int(10)	Primaire
Nom_medecin	Nom du médecin	Varchar(30)	
Prenom_medecin	Prénom du patient.	Varchar(30)	
Tel_medecin	Numéro du téléphone mobile du médecin	Varchar(15)	
Id_service	Identifiant du service dont dépend le médecin	Int(10)	
Distance	Distance du médecin des urgences	Int(10)	
Pseudo_m	Pseudonyme du médecin	Varchar(20)	
Password_m	Mot de passe du médecin	Varchar(20)	

2. Table Message

Attribut	Description	Type	Clé
Id_message	Numéro du message	Int(10)	Primaire
Type_message	Message simple, Alerte	Enum	
Id_emetteur	Identifiant de l'émetteur du message	Int(10)	
Id_recepteur	Identifiant du récepteur du message	Int(10)	
Contenu_message	Texte du message	Text	
Date_message	Date et heure d'envoi du message	Datetime	

3. Table Service

Attribut	Description	Type	Clé
Id_service	Identifiant du service	Int(10)	Primaire
Nom_service	Nom du service	Varchar(10)	

4. Table Receptionniste

Attribut	Description	Type	Clé
Id_recept	Identifiant du réceptionniste	Int(10)	Primaire
Nom_recept	Nom du réceptionniste	Varchar(10)	
Prenom_recept	Prénom du réceptionniste	Varchar(10)	
Tel_recept	Numéro du téléphone mobile du réceptionniste	Varchar(15)	
Pseudo_r	Pseudonyme du réceptionniste	Varchar(20)	
Password_r	Mot de passe du réceptionniste	Varchar(20)	

III.6 Conclusion

Dans ce chapitre nous avons suivi une démarche de modélisation pour développer notre application, en se basant sur l'UML

Nous avons entamé notre démarche par la spécification des besoins et les divers cas d'utilisation, puis l'élaboration des diagrammes de séquence, de classes et la conception de la base de données.

Chapitre IV

Réalisation

IV.1 Introduction

Ce chapitre décrit la mise en œuvre de notre application. La première partie est consacrée à la description des outils utilisés et dans la deuxième partie nous décrirons les détails techniques de l'implémentation.

IV.2 Outils utilisés

Pour le développement de notre application nous avons utilisé divers outils qui sont les suivants :

1. Serveur Web

Pour mettre en place une application Web dynamique, nous avons choisi le serveur web WampServer.

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données. [21]

2. L'environnement de programmation Windev 17

WinDev est un atelier de génie logiciel (AGL) édité par la société française PC SOFT et conçu pour développer des applications, principalement orientées données pour Windows 8, 7, Vista, XP, 2008, 2003, 2000, mais également pour Linux, .Net et Java. Il propose son propre langage, appelé WLangage. La première version de l'AGL est sortie en 1993. Apparenté à WebDev et WinDev Mobile. [22]

3. Le langage WLangage

C'est un langage de programmation de quatrième génération. Ce type de langage de programmation permet d'écrire plus de choses avec moins de lignes de programmes et moins d'erreurs. Facile à apprendre, il permet de décrire certaines opérations de manière non procédurale et permet d'obtenir rapidement des résultats à partir de courts programmes. [23]

4. Windev Mobile 17

WinDev Mobile est un AGL (Atelier de Génie Logiciel) permettant de développer les applications dans tous les domaines.

WinDev est un outil de développement complet qui intègre tous les outils nécessaires au cycle de réalisation d'une application.

Contrairement à d'autres langages traditionnels, il n'est pas nécessaire de chercher et de rajouter des modules pour pouvoir concevoir, tester et installer une application. [24]

5. Le langage HTML (*HyperText Markup Language*)

C'est le langage universel utilisé sur les pages Web lisibles par tous les Navigateurs Web. Ce langage fonctionne suivant l'assemblage et la combinaison de balises permettant de structurer et donner l'apparence voulue aux données textes, images et multimédias suivant la mise en page voulue. [25]

6. Le langage PHP

Appelé initialement Personal Home Page puis devenu Hypertext Preprocessor, PHP s'impose comme un standard dans le monde de la programmation web par ses performances, sa fiabilité, sa souplesse et sa rapidité.

Le langage PHP peut être incorporé dans le code source d'une page HTML mais les scripts programmés en PHP se focalisent principalement sur la partie serveur. Vous pouvez donc faire tout ce qu'un programme CGI¹ est en mesure d'accomplir, comme la collecte de formulaires, la génération de pages dynamiques ou encore envoyer et recevoir des cookies, par exemple.[26]

7. AJAX

AJAX n'est ni une technologie ni un langage de programmation ; AJAX est un concept de programmation Web reposant sur plusieurs technologies comme le JavaScript et le XML – d'où le nom AJAX. A l'heure actuelle, le XML² tend à être délaissé au profit du JSON.

¹ Un script CGI (Common Gateway Interface) est un programme exécuté par le serveur web

² eXtensible Markup Language (ou Langage à balises extensible) est en quelque sorte un langage HTML amélioré

L'idée même d'AJAX est de faire communiquer une page Web avec un serveur Web sans occasionner le rechargement de la page. C'est la raison pour laquelle JavaScript est utilisé, car c'est lui qui va se charger d'établir la connexion entre la page Web et le serveur. [27]

IV.3 Android

IV.3.1 Définition

Android est un système d'exploitation, open source utilisant le noyau Linux, pour téléphone portable de nouvelle génération développé par Google. Celui-ci met à disposition un kit de développement (SDK) basé sur le langage Java.

Android doit son nom à la startup du même nom spécialisée dans le développement d'applications mobiles que Google a rachetée en août 2005, nom qui vient lui-même d'« androïde » qui désigne un robot construit à l'image d'un être humain. [28]

IV.3.2 Le kit de développement (SDK) d'Android

C'est un ensemble complet d'outils de développement incluant un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU³, de la documentation, des exemples de code et des tutoriaux.

L'émulateur que contient SDK permet de simuler les différentes versions d'Android, permettant ainsi aux développeurs de tester leurs applications ou de tester les fonctionnalités d'Android. Le SDK contient plusieurs images en fonction des différentes versions d'Android.[29]

IV.3.3 Composantes d'une application Android

Une application Android est composée d'éléments de base :

1. Activities (Activités en français)

³QEMU est un logiciel libre de machine virtuelle, pouvant émuler un processeur et, plus généralement, une architecture différente si besoin. Il permet d'exécuter un ou plusieurs systèmes d'exploitation via les hyperviseurs KVM et Xen, ou seulement des binaires, dans l'environnement d'un système d'exploitation déjà installé sur la machine.

Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android. Prenant l'exemple d'une application qui liste tous vos fichiers mp3 présents dans votre téléphone, le projet pourrait se décomposer comme ci-dessous :

- une vue pour afficher la liste des mp3 ;
- une activité pour gérer le remplissage et l'affichage de la liste ;

Si l'on veut pouvoir rajouter, supprimer des mp3, on pourrait rajouter d'autres activités.

2. Services

Un service, à la différence d'une activité, ne possède pas de vue mais permet l'exécution d'un algorithme sur un temps indéfini. Il ne s'arrêtera que lorsque la tâche est finie ou que son exécution est arrêtée.

Il peut être lancé à différents moments :

- au démarrage du téléphone ;
- au moment d'un événement (arrivée d'un appel, SMS, mail, etc.) ;
- lancement de votre application ;
- action particulière dans votre application.

3. Broadcast and Intent Receivers

Un *Broadcast Receiver* comme son nom l'indique permet d'écouter ce qui se passe sur le système ou sur votre application et déclencher une action que vous aurez prédéfinie. C'est souvent par ce mécanisme que les services sont lancés.

Les *content providers* servent à accéder à des données depuis votre application. Vous pouvez accéder aux contacts stockés dans le téléphone ; à l'agenda ; aux photos ; ainsi qu'à d'autres données depuis votre application grâce aux *content providers*.

IV.3.4 Cycle de vie d'une application Android

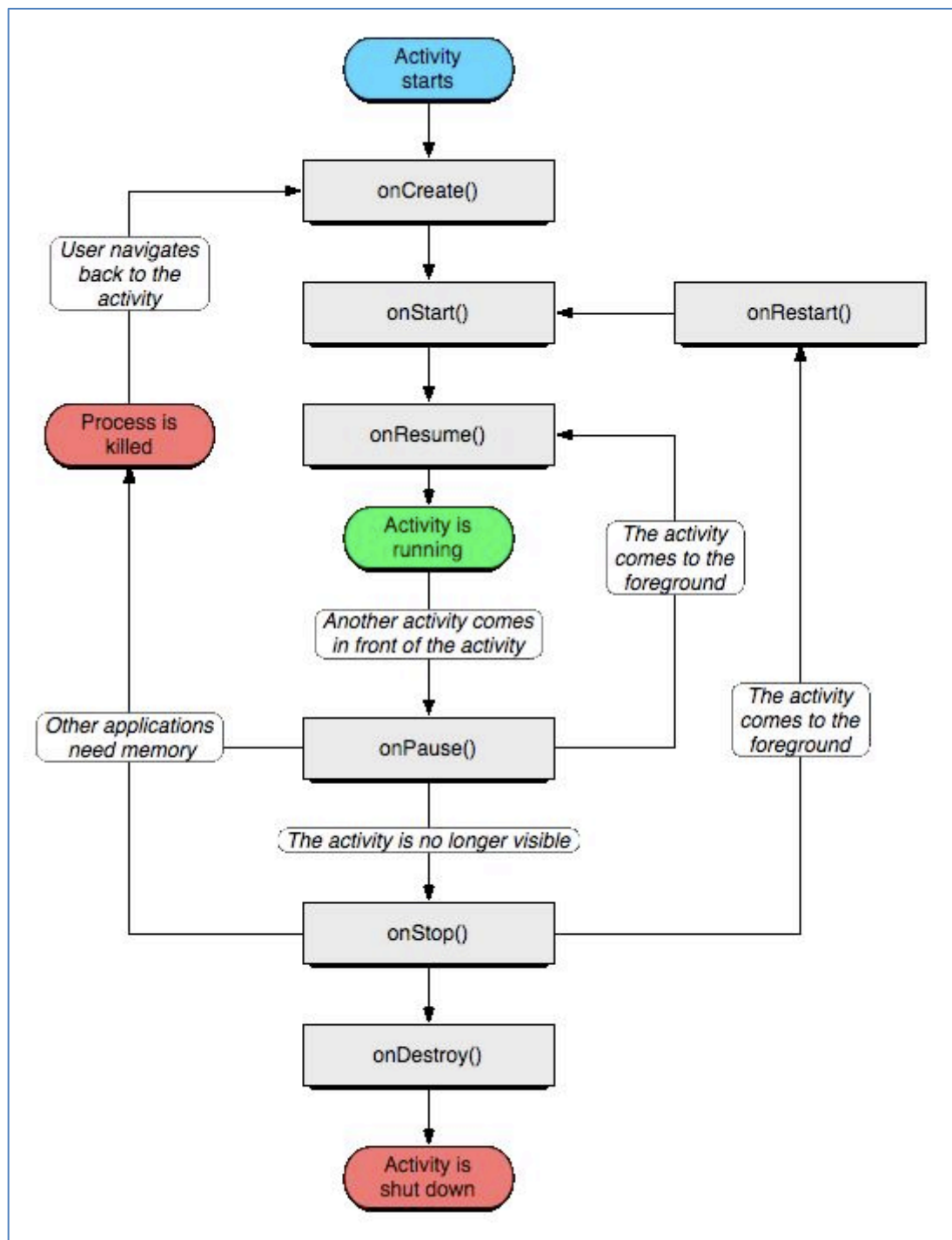


Figure IV.1 Cycle de vie d'une application Android [30]

1. **OnCreate()**

Cette méthode est appelée à la création de votre activité (Activity). Elle sert à initialiser votre activité ainsi que toutes les données nécessaires à cette dernière.

Quand la méthode **OnCreate** est appelée, on lui passe un Bundle en argument. Ce Bundle contient l'état de sauvegarde enregistré lors de la dernière exécution de votre activité.

2. **onStart()**

Cette méthode est appelée dans le cas où votre application est en arrière-plan et qu'elle repasse en avant-plan.

Si votre activité ne peut pas aller en avant-plan quelle que soit la raison, l'activité sera transférée à **OnStop()**.

3. **onResume()**

Cette méthode est appelée après **OnStart** (au moment où votre application repasse en avant-plan).

OnResume() est aussi appelée quand votre application passe en arrière-plan à cause d'une autre application.

4. **onPause()**

Appelée juste avant qu'une autre activité que la vôtre passe en **OnResume**. À ce stade, votre activité n'a plus accès à l'écran, vous devez arrêter de faire toute action en rapport avec l'interaction utilisateur. Vous pouvez par contre continuer à exécuter des algorithmes nécessaires mais qui ne consomment pas trop de CPU.

5. onStop()

Appelée quand votre activité n'est plus visible quelle que soit la raison.

6. onDestroy()

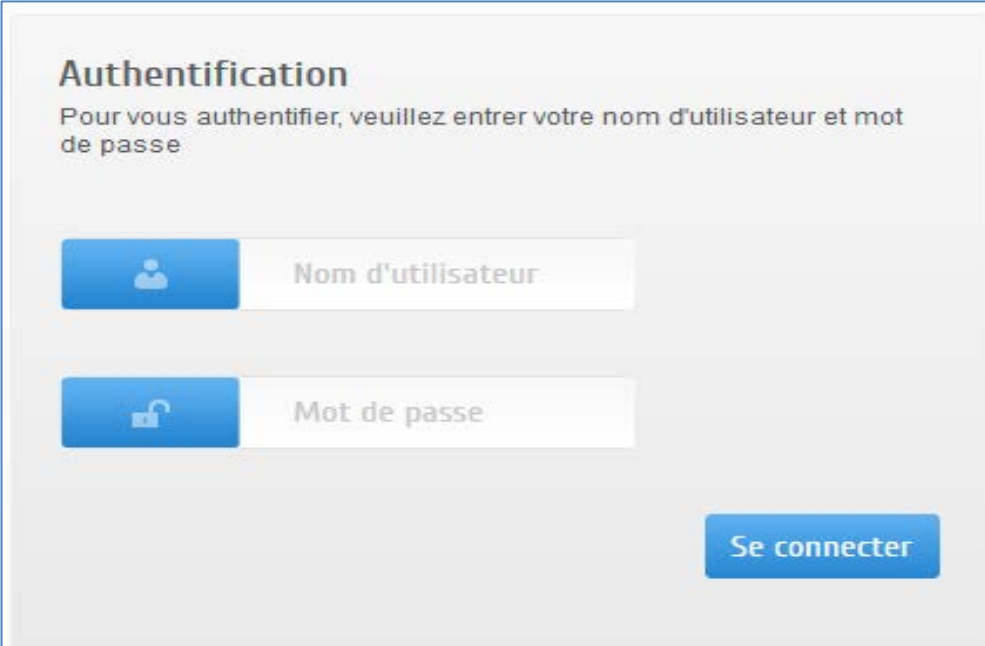
Elle est appelée quand l'application est totalement fermée (Processus terminé).

IV.4. Description des interfaces de l'application Server

Nous décrivons dans cette section les différentes interfaces de notre application côté Server.

IV.4.1 Description de l'interface d'authentification

Cette première capture présente l'interface d'authentification dans laquelle on doit entrer le nom d'utilisateur et le mot de passe pour commencer à utiliser notre application. Cette interface constitue la fenêtre d'accueil de notre application. A travers celle-ci l'utilisateur pourra s'authentifier pour utiliser l'application. Cette étape met en valeur l'aspect sécurité : elle permet de vérifier l'existence du compte utilisateur, le mot de passe correspondant et les droits et privilèges qui lui sont attribués.



Authentification
Pour vous authentifier, veuillez entrer votre nom d'utilisateur et mot de passe

Se connecter

Figure IV.2 Interface d'authentification

Dans notre application, nous avons deux types d'utilisateurs : les administrateurs et les réceptionnistes. Par conséquent, nous avons prévu un niveau d'accès pour chacun d'entre eux.

Les droits des réceptionnistes se résument à la consultation de la liste des médecins, l'envoi des messages et alertes et la lecture des messages.

Les administrateurs ont tous les droits. Ce qui se résume à :

- L'ajout, la modification, la suppression des services.
- L'ajout, la modification, la suppression des médecins.
- L'ajout, la modification, la suppression des réceptionnistes.
- L'affectation de médecins à des services.
- La consultation de l'historique des messages / alertes émise par les réceptionnistes.

IV.4.2 Description de l'interface principale

Dans l'interface principale, vous remarquerez que les médecins sont triés par service et ordonné du plus proche au plus lointain. Les médecins connectés sont affichés avec une lampe verte à gauche. Les médecins non connectés ou hors de portée sont précédés d'une lampe rouge.

Cette interface nous permet d'avoir les positions et les réponses des médecins avec une fréquence d'une seconde qui nous évite le rafraîchissement manuel de notre navigateur pour avoir nos mises à jour.

Pour envoyer une alerte il suffit de cliquer sur le médecin voulu (généralement le plus proche et suivant la spécialité du service voulu) et de cliquer sur l'une des deux alertes mise à disposition.

- Alerte service : indique au médecin sélectionné qu'il est demandé en urgence dans son service.
- Alerte urgence : indique au médecin sélectionné qu'il est demandé en urgence au service des urgences de l'hôpital.

Une fois l'alerte émise un compte à rebours est lancé et à la fin de ce dernier s'il n'y a pas de réponse à l'alerte un message système demandant d'envoyer l'alerte à un autre médecin devra s'afficher.

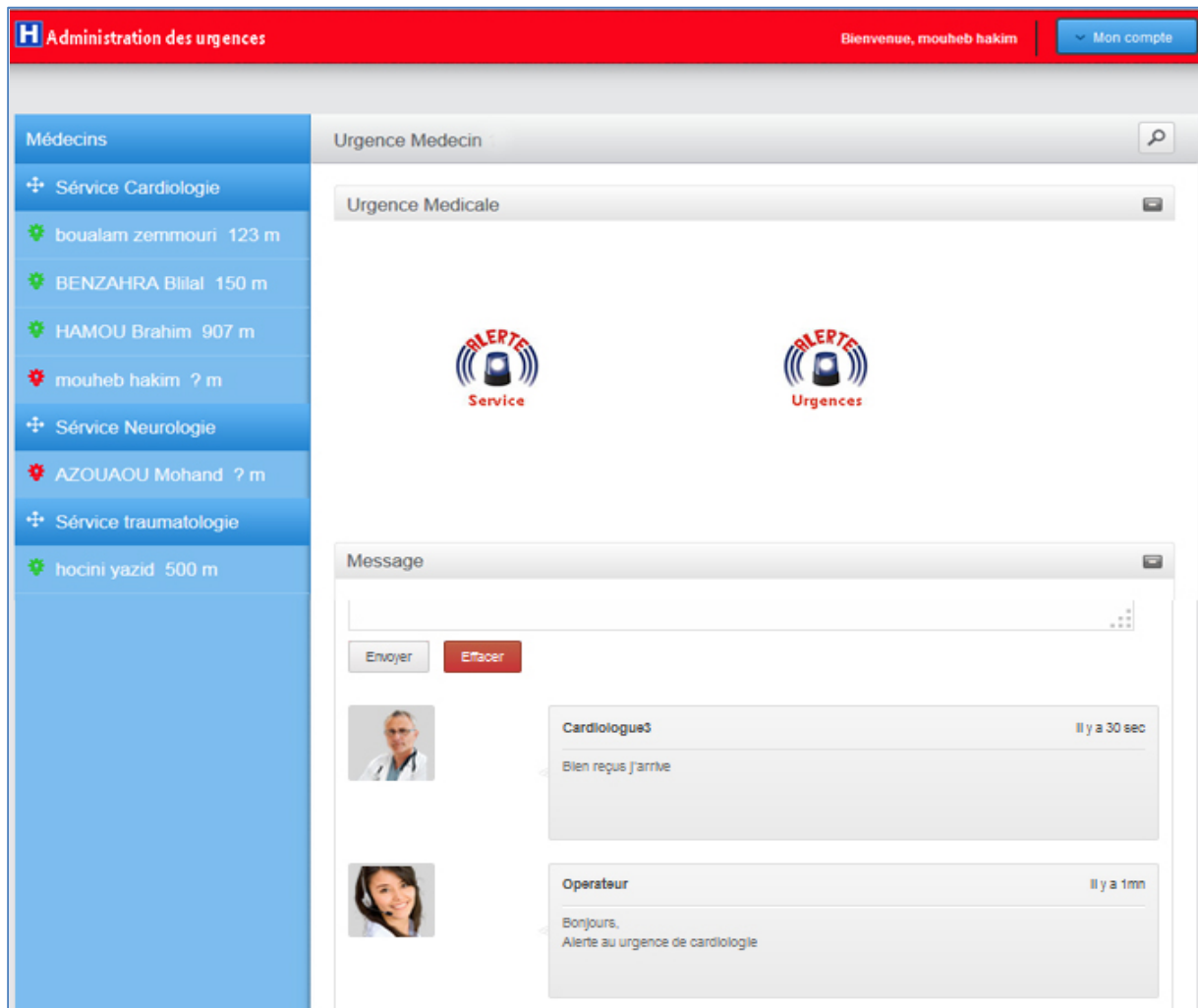


Figure IV.3 Interface principale de l'application

IV.4.3 Description de l'interface de l'administrateur

Comme l'administrateur a tous les droits donc, en plus des informations affichés aux réceptionnistes, il a la possibilité d'ajouter, de supprimer, de modifier des informations concernant les services, les médecins et les réceptionnistes. La figure IV.4 montre l'interface de l'administrateur.

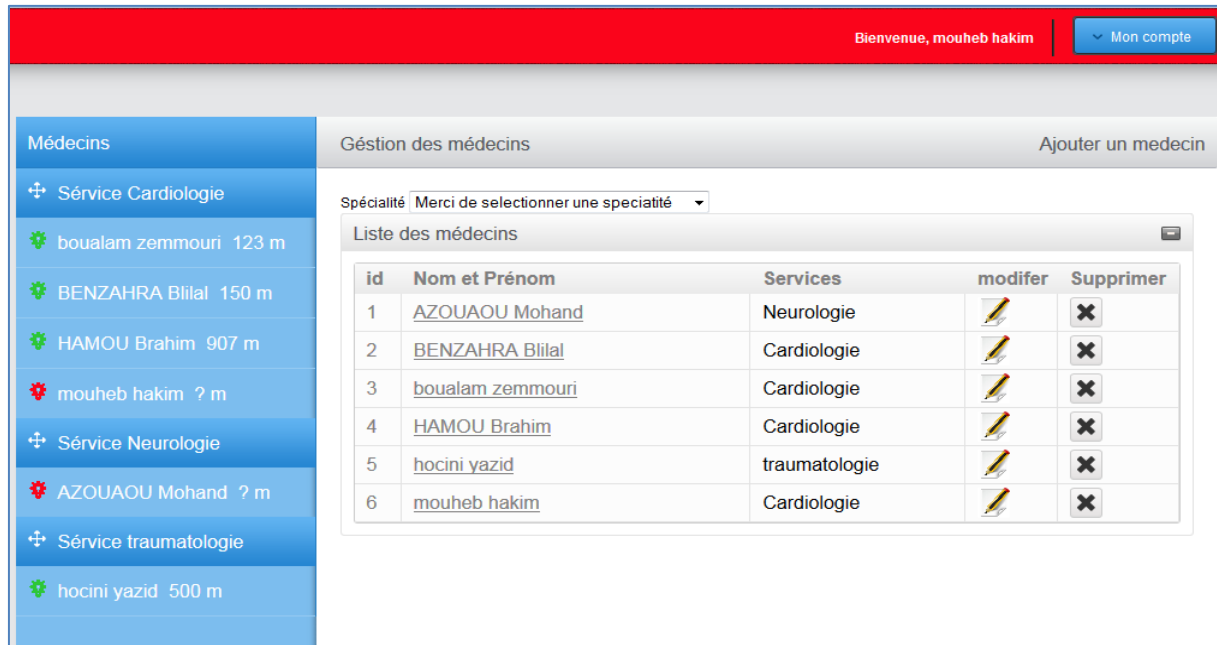


Figure IV.4 Interface Gestion de médecins

IV.4.3.1 Formulaire d'ajout de médecin

La figure IV.5 représente le formulaire d'ajout de médecin.

H Administration des urgences

Bienvenue, mouheb hakim | Mon compte

Médecins

Ajouter un medecin

Formulaire d'ajout

Nom d'utilisateur:

Mot de passe:

Confirmez votre mot de passe:

Nom:

Prénom:

Telephone mob:

Service:

Enregistrer Effacer

Sérvices

- Sérvices Cardiologie
- boualam zemmouri 123 m
- BENZAHA Blilal 150 m
- HAMOU Brahim 907 m
- mouheb hakim ? m
- Sérvices Neurologie
- AZOUAOU Mohand ? m
- Sérvices traumatologie
- hocini yazid 500 m

Figure IV.5 Interface création d'un médecin

IV.4.3.2 Interface de gestion de service

La figure IV.6 représente le formulaire d'ajout de service au niveau de l'espace d'administrateur.

id	Services	modifier	Supprimer
1	Cardiologie		
2	Neurologie		
3	traumatologie		

Figure IV.6 Interface gestion des services

IV.4.3.3 Formulaire d'ajout de service

La figure IV.7 représente le formulaire d'ajout de service.

Figure IV.7 Interface création d'un service

IV.5. Description des interfaces de l'application Client (Android)

IV.5.1 Description de l'interface d'authentification

Cette interface représente l'interface d'authentification réservée aux médecins dans laquelle ces derniers doivent entrer leur nom d'utilisateur et le mot de passe pour commencer à utiliser notre application. Cette interface constitue la fenêtre d'accueil de notre application.

Cette étape met en valeur l'aspect sécurité et permettra à la partie serveur d'identifier les médecins sur le réseau.



Figure IV.8 Interface d'authentification de la partie cliente

IV.5.2 Description de l'interface de disponibilité

Après authentification les médecins devront indiquer leur état de disponibilité à savoir disponible ou pas. Ceci est fait pour optimiser notre lancement d'alerte car on n'envoie pas d'alerte a un médecin déjà occupé par exemple un médecin en plein intervention chirurgical.

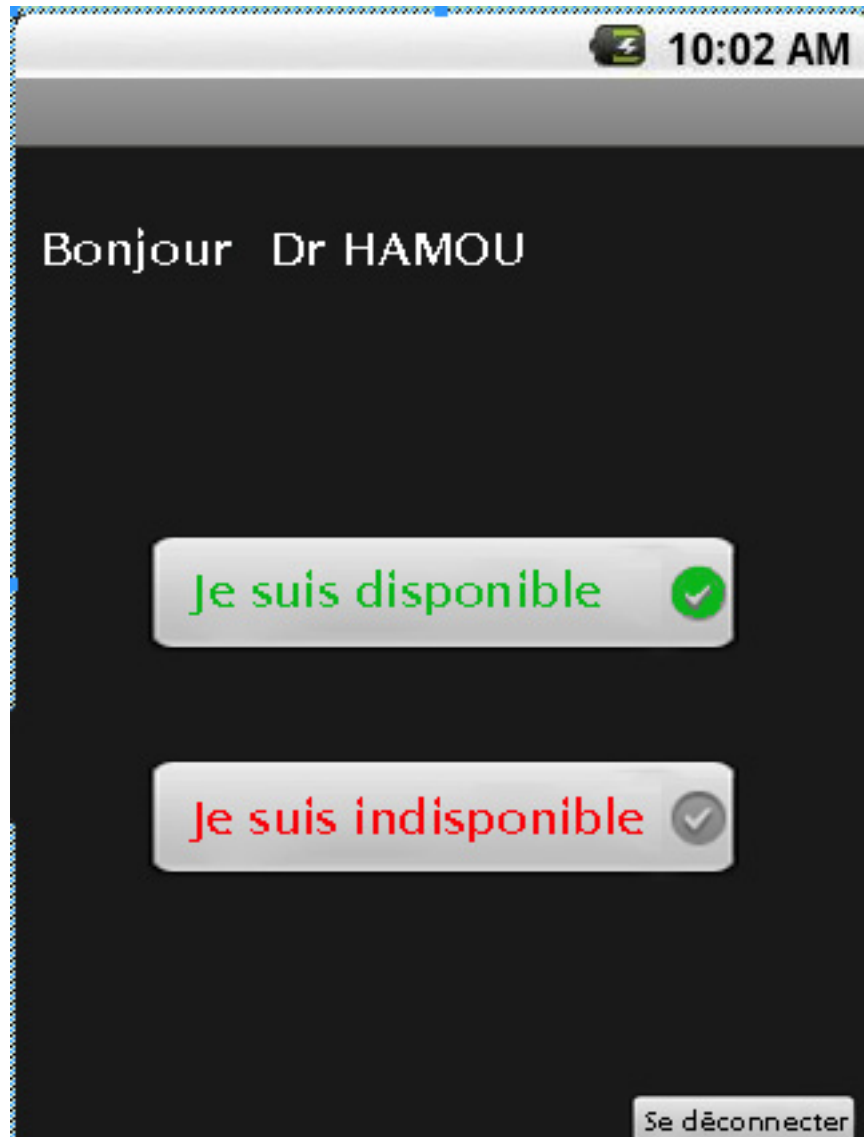


Figure IV.9 Interface de mise à disposition du médecin

IV.5.3 Description de l'interface de réception d'alerte

Cette interface permettra aux médecins de recevoir leurs alertes néanmoins il aura un laps de temps pour y répondre dans quel cas l'alerte sera transmise un autre médecin et bien sure ne sera plus affiché chez le premier

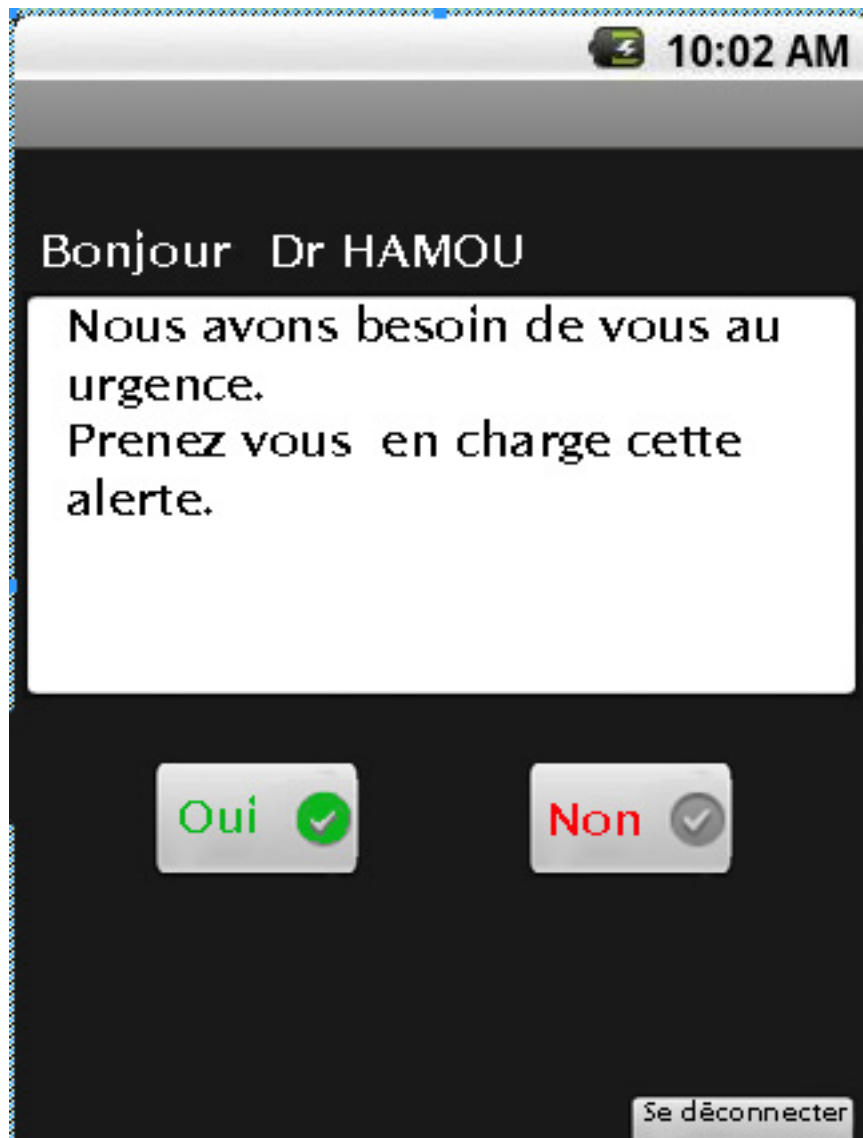


Figure IV.10 Interface de réception et de réponse a une alerte d'alerte

IV.6 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre de notre application en décrivant les différentes interfaces des différents acteurs au niveau des services des urgences.

Nous avons cité les outils de développement de notre application, puis nous avons présenté les interfaces de la partie server. Enfin, nous avons décrit la partie client avec une brève explication à chaque interface.

Conclusion générale

L'objectif de ce travail est de mettre en œuvre une application de gestion des urgences d'un hôpital dont l'objectif principal est de gérer efficacement les alertes destinées aux médecins en fonction du cas d'urgence, de la disponibilité et/ou la proximité du médecin de l'hôpital.

La solution choisie est de repérer les médecins via leurs Smartphones dotés de GPS. Cette solution peut s'avérer très utile dans des situations nécessitant une grande mobilisation du personnel médical où même des médecins se trouvant à des kilomètres de l'hôpital peuvent être appelés en renfort.

Toutefois, cette solution peut présenter des imperfections du fait des difficultés de localiser des personnes, via leurs Smartphones équipés de GPS, si celles-ci se trouvent indoor.

Et pour cela, et comme perspectives, nous suggérons de combiner cette solution avec la localisation par WiFi et par GPRS.

Il serait aussi intéressant d'étendre le système au personnel paramédical.

Bibliographie

- [1] Tout sur les Réseaux et Internet, J-F. PILLOU et F. LEMAINQUE, Dunod, Paris, 2012
- [2] www.commentcamarche.net
- [3] http://webia.lip6.fr/~lepape/ens/ELI5-resmob/cours/cours1_reseauxcellulaires.pdf
- [4] Protocoles de routage réactifs pour l'optimisation de bande passante et la garantie de délai dans les réseaux ad hoc mobiles, David Espès, Thèse de Doctorat en informatique, Université Toulouse III - Paul Sabatier, novembre 2008
- [5] Optimisation des applications de streaming peer to peer pour des réseaux ad hoc mobiles, Mehdi NAFA, Thèse de Doctorat en informatique, Université d'Evry-Val-d'Essonne France, 2009
- [6] DZ-MAODV : Nouveau protocole de routage multicast pour les réseaux ad hoc mobiles basé sur les zones denses, Abdelhamid ZEBDI, Mémoire de maîtrise en mathématiques et Informatique appliquées, Université du Québec à Trois-Rivières, 2006
- [7] Sécurité du routage dans les réseaux ad hoc Mobile, Khadidja AYAD, thèse de Magister, Ecole nationale Supérieure en Informatique (ESI) Alger, 2012
- [8] L'effet de la mobilité sur les protocoles de routages dans les réseaux Ad Hoc, D BE-CAYE et T. CHEICKNA, Mémoire d'Ingénieur en Informatique UMMTO 2007
- [9] Contrôle intelligent de flux capable de s'adapter à l'état d'un Manet, Malika BELKADI, Thèse de Doctorat en Informatique, Département d'Informatique UMMTO, 2012
- [10] Wikipedia: [<http://fr.wikipedia.org/wiki/G%C3%A9olocalisation>]
- [11] http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_positionnement_par_satellites
- [12] <http://www.gps.gov>
- [13] www.esa.int
- [14] <http://eduscol.education.fr/orbito/system/navstar/gps1.htm>
- [15] <http://fr.wikipedia.org/wiki/G%C3%A9olocalisation>
- [16] www.rfit-tech.com/tracabilite/architecture-rfid/identification-automatique/rfid
- [17] Kevin Vuillard, Laurent Duplat, Laurène Hardouin, Cyril Ressay et Jimmy Sourdin, Géolocalisation et Smartphones dans le domaine culturel et patrimonial, livre blanc, 2011.
- [18] [http://fr.wikipedia.org/wiki/UML_\(informatique\)](http://fr.wikipedia.org/wiki/UML_(informatique))

- [19] UML 2 en action: de l'analyse des besoins à la conception, Pascal Roques, Franck Vallée, Editions Eyrolles, 2007.
- [20] <http://laurent-audibert.developpez.com/Cours-UML/>
- [21] <http://www.wampserver.com/>
- [22] <http://fr.wikipedia.org/wiki/WinDev>
- [23] <http://fr.wikipedia.org/wiki/WLangage>
- [24] www.pcsoft.fr
- [25] <http://www.clashinfo.com/>
- [26] www.01net.com
- [27] <http://fr.openclassrooms.com/>
- [28] <http://android.zeblogger.com/generalites/>
- [29] http://fr.wikipedia.org/wiki/Android_SDK
- [30] <http://nbenbourahla.developpez.com/tutoriels/android/introduction-programmation-android/>

Annexes

```
<?PHP

session_start();

$moiChri="N";

/* Coordonnés hopital
$long_serv=36.7087698;
$lat_serv=4.0544486;*/
require("ajax_sql.php");
$long_serv=36.696049;
$lat_serv=4.055752;

//----- Calcul de la distante entre 2 point GPS
function get_distance_m($lat1, $lng1, $lat2, $lng2) {
    $earth_radius = 6378137; // Terre = sphère de 6378km de rayon
    $rlo1 = deg2rad($lng1);
    $rla1 = deg2rad($lat1);
    $rlo2 = deg2rad($lng2);
    $rla2 = deg2rad($lat2);
    $dlo = ($rlo2 - $rlo1) / 2;
    $dla = ($rla2 - $rla1) / 2;
    $a = (sin($dla) * sin($dla)) + cos($rla1) * cos($rla2) * (sin($dlo) * sin($dlo
));
    $d = 2 * atan2(sqrt($a), sqrt(1 - $a));
    return ($earth_radius * $d);
}

$pseudo = (isset($_SESSION['pseudo'])) ? secur($_SESSION['pseudo']) : "";

$ActuPseudoID = (isset($_SESSION['ActuPseudoID'])) ?
$_SESSION['ActuPseudoID'] : "";

$action = (isset($_POST['action'])) ? secur($_POST['action']) : "";

$message = (isset($_POST['message'])) ? secur($_POST['message']) : "";
```

```

$liste="";

if($pseudo=="" || $action=="") {

    $liste="Vous devez être connect, pour utiliser le tchat";

}

else {

    if($action=="Qui_En_Ligne") {

        //=====Si appel AJAX "Qui_En_Ligne" =====

        $now = time();

        $service = $bdd->prepare("SELECT * FROM service ORDER BY nom_serv
ASC ");

        $service->execute();

        while ($row_serv = $service->fetch())

        {

            $result = $bdd->prepare("SELECT * FROM medecin WHERE ser-
vice='".$row_serv['id_serv']."' and `etat-pre`='y' ORDER BY localisation ASC ");

            $result->execute();

            $liste.='<li>

                <a href="#"><span class="four-prong icon"> Service
'.$row_serv['nom_serv'].'</span></a>

                <ul class="acitem">';

            while ($row = $result->fetch())

            {

                $localisation=$row['localisation'];

                /* $localisation= round(get_distance_m($lat_serv, $long_serv,
$row['lat'], $row['long']) , 3).' m';*/

                if ($row['etat-pre']=='y'){ $img_etat='steps_v';}

                $liste.='<li><a href="table.php?id_med='.$row['id_med'].'"><span
class="'.$img_etat.' icon">'.$row['nom'].'&nbsp;'.$row['prenom'].

                '&nbsp;&nbsp; '.$localisation ;

            $liste.=' m</span></a></li>';

```

```

    }

    $result->closeCursor();

    $result2 = $bdd->prepare("SELECT * FROM medecin WHERE
service='".$row_serv['id_serv']."' and `etat-pre`='n' ORDER BY localisation ASC ");

    $result2->execute();

    while ($row = $result2->fetch())
    {

        $localisation=$row['localisation'];

        /* $localisation= round(get_distance_m($lat_serv, $long_serv,
$row['lat'], $row['long']) , 3).' m';*/

        if ($row['etat-pre']=='n') { $img_etat='steps_r';}

        $liste.= '<li><a href="table.php?id='.$row['id_med'].'"><span
class="'.$img_etat.' icon">'.$row['nom'].'&nbsp;'.$row['prenom'].

        '&nbsp;&nbsp;&nbsp;';

        $liste.='? m</span></a></li>';

    }

    $liste.='</ul></li>';

    $result2->closeCursor();}

    $service->closeCursor();

}

if($action=="Ajout") {

    //=====Si appel AJAX "Ajout" par un visiteur d'un nouveau message =====

    $message=secur($message);

    $result = $bdd->prepare("INSERT INTO message

    ( pseudo ,pseudoid, message , date ) VALUES (:Xpseudo,:Xpseudoid, :Xmessage,

    ".time().")");

```

```

$result->execute(array(':Xpseudo' => $pseudo, ':Xpseudoid' =>
$_GET['id'], ':Xmessage' => $message));

$result->closeCursor();

$now = time();

$result = $bdd->prepare("UPDATE utilisateur SET date = :Lenow WHERE id_user
= :Avat");

$result->execute(array(':Lenow' => $now, ':Avat' => $ActuPseudoID ));
}

if($action=="liste_medecin") {

//=====si appel AJAX "liste medecin" =====

$now = time();

$medecin = $bdd->prepare("SELECT * FROM medecin ORDER BY nom ASC
");

$medecin->execute();

$liste.='<table class="display" >

<thead>

<tr>

<th class="th_chexbox">id</th>

<th class="th_title">Nom et Prénom</th>

<th class="th_date">Services</th>

<th class="th_chexbox">modifier </th>

<th class="th_chexbox">Supprimer </th>

</tr>

</thead>';

$inc=0;

while ($row_medecin = $medecin->fetch())

{

$inc=$inc+1;

$liste.= '<tbody>

```

```

        <tr class="item">
            <td>'.$inc.'</td>

            <td class="subject"><a
href="#">'.$row_medecin["nom"].'&nbsp;'.$row_medecin["prenom"].'</a></td>

            <td><a href="#">';

        $service = $bdd->prepare("SELECT * FROM service where
id_serv='".$row_medecin['service']."' ORDER BY nom_serv ASC ");

        $service->execute();

        while ($row_serv = $service->fetch())

        { $service_med=$row_serv['nom_serv']; } $service->closeCursor();

        $liste.=$service_med.'</a></td>

            <td class="action"><a
href="mod.php?id_med='.$row_medecin["id_med"].'"></a></td>

            <td class="action"><a
href="supp.php?id_med='.$row_medecin["id_med"].'"></a></td>

        </tr>

    </tbody>'; }

    $medecin->closeCursor();}

//=====Début appel AJAX "liste des services" =====

if($action=="liste_service") {

    $now = time();

    $service = $bdd->prepare("SELECT * FROM service ORDER BY nom_serv
ASC ");

    $service->execute();

    $liste.='<table class="display" >

```

```

<thead>

<tr>

<th class="th_chexbox">id</th>

<th class="th_date">Services</th>

<th class="th_chexbox">modifier </th>

<th class="th_chexbox">Supprimer </th>

</tr>

</thead>;

$inc=0;

while ($row_service = $service->fetch())
{

    $inc=$inc+1;

    $liste.= '<tbody>

<tr class="item">

<td>'.$inc.'</td>

<td class="subject"><a href="#">'.$row_service["nom_serv"].'</a></td>

<td class="action"><a
href="mod.php?id_serv='.$row_service["id_serv"].'"></a></td>

<td class="action"><a
href="supp.php?id_serv='.$row_service["id_serv"].'"></a></td>

</tr>

</tbody>'; }

    $service->closeCursor();}

//===== Fin appel AJAX " liste des services " =====

//===== Début appel AJAX "Affiche_message" =====

if($action=="Affiche_message")    {

```



```

$lastid = 0;

$result = $bdd->prepare("SELECT * FROM message ORDER BY date DESC ");
$result->execute();

$liste .=";while ($row = $result->fetch())
{
    $lim=10000+$row['id_emmeteur']; // =====l'image (avatar) est
    ajax/tchat10003.jpg si l'ID du pseudo est 3 assez simpliste mais vous ferez mieux ===
    $LaDate=date("d/m/Y H:i:s",$row['date']);

    if($row['pseudo']=='opérateur')
    { $image='doc.png';
    }else
    { $image='standard.png';}

    $liste .=<div class="message-list">

        <div class="message">

            <div class="avatar"></div>

            <div class="message-content">

                <div class="message-info">

                    <div class="message-author"><p>'.$row['pseudo'].'</p></div>

                    <div class="message-date"><p>'.$LaDate.'</p></div>

                    <div class="message-border-head"></div>

                    <div class="message-text">

                        <p>'.$row['message'].'</p></div>

                    </div>

                </div>

            <div class="clear"></div>

        </div>';}

```

```
$result->closeCursor();

$now = time();

$result = $bdd->prepare("UPDATE utilisateur SET date = :Lenow WHERE
id_user = :Avat");

$result->execute(array(':Lenow' => $now, ':Avat' => $ActuPseudoID ));

}

} //===== fin appel AJAX "Affiche_message" =====

echo $liste;

?>
```

```
<?php
try {
    $bdd = new PDO('mysql:host=localhost;dbname=hopital', 'root', '');
    $bdd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $bdd->query("SET NAMES 'utf8'");
}
catch(Exception $e) {
    die('Erreur : '.$e->getMessage());
}

header('Content-type: text/html; charset=UTF-8');

function secur($a)
{
    $a=str_replace("<", "", $a);
    $a=str_replace(">", "", $a);
    $a=str_replace("&&", "", $a);
    $a=str_replace("||", "", $a);
    $a=str_replace("◆", " ", $a);
    $a=str_replace("\\", "", $a);
    $a=str_replace("/", "", $a);
    $a=str_replace(" ", " ", $a);
    $a=str_replace(" AND ", "", $a);
    $a=str_replace(" OR ", "", $a);
    $a=str_replace(" and ", "", $a);
    $a=str_replace(" or ", "", $a);
    $a=str_replace("◆", "OE", $a);
    $a=str_replace("◆", "oe", $a);
    $a=str_replace("\0", "", $a);
    $a=str_replace("\0", "", $a);
    $a=str_replace("\n\n", "\n", $a);
```

```
$a=str_replace("\r","",$a);  
$a=str_replace("","",$a);  
return $a;  
}  
?>
```

```
<?php
session_start();

$pseudo = (isset($_SESSION['pseudo'])) ? $_SESSION['pseudo'] : "";
if($pseudo=="") header("location:index.php");
header('Content-type: text/html; charset=UTF-8');
$moiChri="O";
require("ajax_sql.php");
?>

<script type="text/javascript">

var timer = setInterval(Affiche_Tchat,5000);
var ctimer = setInterval(Qui_En_Ligne,10000);
function maFonctionAjax(message, pseudo,action)
{
    var MonAjax;

    if (window.XMLHttpRequest){ MonAjax = new XMLHttpRequest();}
    else if (window.ActiveXObject) {MonAjax = new ActiveX-
Object('Microsoft.XMLHTTP');}
    else {
        alert("Votre navigateur n'est pas adapté pour faire des requêtes AJAX...");
        MonAjax = false;  }
    MonAjax.open('POST',"ajax_tchat_pdo.php",true);
    MonAjax.onreadystatechange = function()
    {
        if (MonAjax.readyState == 4 && MonAjax.status == 200)
        {
            if(action=="Affiche_Tchat") { docu-
ment.getElementById('LeRetourL').innerHTML = MonAjax.responseText;}

            if(action=="Qui_En_Ligne") { docu-
ment.getElementById('LeRetour').innerHTML = MonAjax.responseText;}

            if(action=="Ajout") { document.LeForm.message.value="";}

            setTimeout("image()", 2000);
```

```
    }
    else{
        document.getElementById('load').style.visibility="visible";
    }
}

MonAjax.setRequestHeader('Content-type','application/x-www-form-urlencoded');
MonAjax.send('message='+message+'&pseudo='+pseudo+'&action='+action);
}

function Affiche_Tchat(){
<?PHP
echo "pseudo='".$pseudo."'";
?>

maFonctionAjax("", pseudo,"Affiche_Tchat");
}

function Qui_En_Ligne(){
<?PHP
echo "pseudo='".$pseudo."'";
?>

maFonctionAjax("", pseudo,"Qui_En_Ligne");
}

function image()
{
document.getElementById('LeRetourL').style.visibility="visible";
document.getElementById('load').style.visibility="hidden";
}

</script>
```