

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de projets informatiques**

Présenté par
Sylia BELKADI

Thème

Réalisation d'une application mobile pour la localisation de personnel médical dans une ville

Mémoire soutenu publiquement le 03/10/2016 devant le jury composé de :

Président : Mme Rachida AOUDJIT

Encadreur : Mme Karima OUKFIF

Examineur : Mme Siham BOUAMRA

Examineur : Mme Rabiha HADAoui

Remerciements

Nous remercions le Bon dieu de nous avoir donné le courage et la volonté pour la réalisation de notre projet.

Nous tenons à remercier notre promotrice Mme OUKFIF pour sa disponibilité, son aide, ses conseils précieux, ses critiques constructives, ses explications et ses suggestions pertinentes.

Nous tenons à exprimer toute notre gratitude aux membres du jury pour avoir accepté d'évaluer et de juger notre travail.

Et enfin on remercie tous ceux qui ont contribué de loin ou de près pour la réalisation de notre projet.

Dédicaces

Je dédie notre travail

*A mes chers parents pour tous les sacrifices qu'ils ont
faits tout au long de mes études.*

A mes deux sœurs Aldjia, Ouerdia

*A mes deux frères Mehana et Rayanne que j'adore
enormement*

*A toute ma famille en particulier mes
grand_parents et mes oncles*

*A mon oncle Slimane et ma tante Aldjia qui a été toujours
là pour me soutenir moralement*

*A tous mes amis qui m'ont soutenus
en particulier sadia et salima*

*A toutes les personnes qui ont participé de près ou de
loin à l'aboutissement de ce travail*

Sydia

Sommaire

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

Chapitre I : Les systèmes d'exploitation mobiles

I. Introduction.....	3
II. Système d'exploitation des Smartphones.....	3
II.1. iPhone OS	3
II.2. BlackBerry	4
II.3. Symbian	5
II.4. Windows Mobile	6
II.5. Android OS	7
II.2. Comparaison entre les différents Systèmes d'exploitation mobiles	7
III. Présentation de la plateforme Android	9
III.1. Architecture d'Android.....	9
III.1.1. Application	9
III.1.2. Framwork Applicatif	9
III.1.3. Les libraries	9
III.1.4. Environnement d'exécution Android	10
III.1.5. Le noyau linux.....	11
III.2. Les concepts d'une application Android	11
III.2.1. L'activité (Activity)	11
III.2.1.1. Etat d'une activité	13
III.2.1.2. Cycle de vie d'une activité	13
III.2.2. Les Services.....	14

Sommaire

III.2.3. Content Provider	15
III.2.4. Broadcast Receiver.....	15
III.2.5. Intent.....	15
IV. Historique et versions d'android.....	16
IV.1. Open Handset Alliance	16
IV.2. Bugdroïde.....	16
IV.3. Versions d'Android	17
V.Différents équipements utilisant Android.....	17
V.1.Les Smartphones	17
V.2. Les Tablettes.....	18
V.3. Les télévisions	18
V.4. Les télévisions	18
VI. Conclusion	19

Chapitre II : Analyse et conception

I. Introduction.....	20
II. La problématique et la solution proposée.....	20
III. Présentation d'UML.....	21
III.1. Définition.....	21
III.2. Les diagrammes UML.....	21
III.3. La démarche de modélisation.....	21
IV. Analyse	22
IV.1. Identification des acteurs	22
IV.2. Diagramme de contexte	23
IV.3. Les cas d'utilisation	24

Sommaire

IV.4. Description des cas d'utilisation avec des scénarios	26
IV.4.1. Description du cas d'utilisation « Rechercher personnel médical ou paramédical »	27
IV.4.2. Description du cas d'utilisation « Afficher».....	28
IV.4.3. Description du cas d'utilisation « Prendre RDV»	29
IV.4.4. Description du cas d'utilisation « Appeler personnel médical ou paramédical »	30
IV.4.5. Description du cas d'utilisation «Authentification Administrateur».....	31
IV.4.6. Description du cas d'utilisation « Ajouter personnel médical ou paramédical coté administrateur»	31
IV.4.7. Description du cas d'utilisation « Ajouter RDV coté médical ou paramédical »	32
IV.5. Conception	32
IV.5.1. Les diagrammes de séquences	32
IV.5.1.1. Diagramme de séquence « Rechercher personnel médical ou paramédical »	33
IV.5.1.2. Diagramme de séquence «Appeler le personnel».....	34
IV.5.1.3. diagramme de séquence « Prendre RDV»	35
IV.5.1.4. Diagramme de séquence « Authentification administrateur».....	36
IV.5.1.5. Diagramme de séquence «Ajouter un personnel médical coté Administrateur».....	37
IV.5.2. Le diagramme d'activité	38
IV.5.2.1. Le diagramme d'activité «Authentification Administrateur»	38
IV.5.2.2. Diagramme de séquence « Afficher profil personnel médical »	39

Sommaire

IV.5.3.Diagramme de classe	40
V. Les tables	41
VI. conclusion.....	43

Chapitre III : Implémentation et test

I. Introduction	44
II. Environnement de travail.....	44
II.1. Matériels utilisés.....	44
II.2. Environnement logiciel	44
II.2.1.Android Studio	44
II.2.2. Software Development Kit (SDK)	45
II.2.3. Easy PHP	46
II.2.4. Le serveur MySQL.....	46
II.3. Les langages utilisés	46
II.3.1. Le langage JAVA	46
II.3.2. XML	47
II.3.3. Le CSS.....	47
II.3.4. PHP	47
II.4. Choix de l'architecture de l'application	48
II.4.1. Protocole de communication	49

Sommaire

II.4.2. Format de données communiquées	49
III. Quelques interfaces	50
III.1. Interface Authentification d'administrateur.....	50
III.2. Interface ajout personnel.....	50
III.3. Interface liste personnel	51
III.4. Interface ajout R D V	51
III.5. Interface d'accueil(android).....	52
III.6. Interface de liste de personnels (coté android)	53
III.7. Interface envoyé un mail.....	54
III.8. Interface de la liste des rendez_vous	55
IV. conclusion.....	56

Table des illustrations

Les figures:

Chapitre I: Les systèmes d'exploitation mobiles

Figure I.1: Icône représentant le système iOS	4
Figure I.2: Exemple d'un téléphone utilisant BlackBerry OS.....	5
Figure I.3: Exemple d'un téléphone sous OS Symbian	6
Figure I.4: Icône représentant le système Windows Mobile	7
Figure I.5: Performances des Systèmes d'Exploitation mobiles	8
Figure I.6: Architecture d'un système Android	10
Figure I.7: Différence entre la JVM et Dalvik	11
Figure I.8: Cycle de vie d'une activité.....	14
Figure I.9: Versions d'Android.....	17

Chapitre II: Analyse et conception

Figure II.1: Diagramme de contexte	24
Figure II.2: Diagramme de cas d'utilisation pour l'application mobile	25
Figure II.3: Diagramme de cas d'utilisation pour la partie web	26
Figure II.4: Diagramme de cas d'utilisation «Rechercher personnel médical ou paramédical».....	27
Figure II.5: Diagramme de cas d'utilisation «Afficher profil».....	28
Figure II.6: Diagramme de cas d'utilisation «prendre RDV»	29
Figure II.7: Diagramme de cas d'utilisation «Appeler personnel médical ou paramédical».....	30
Figure II.8: Diagramme de séquence «Rechercher personnel médical ou paramédical»	33
Figure II.9: Diagramme de séquence «Appeler le personnel».....	34
Figure II.10: Diagramme de séquence «Prendre RDV»	35
Figure II.11: Diagramme de séquence «Authentification administrateur»	36
Figure II.12: Diagramme de séquence «Ajouter un personnel coté administrateur»	37
Figure II.13: Diagramme d'activité «Authentification administrateur»	38
Figure II.14: Diagramme d'activité «Afficher profil du personnel»	39
Figure II.15: Diagramme de classe globale.....	39

Chapitre III: Implémentation et test

Figure III.1: Interface de l'environnement Android Studio 2.1.3	44
Figure III.2: Téléphone virtuel d'Android.....	45
Figure III.3: La connexion à distance (Andrid/MySQL)	48
Figure III.4: Interface Authentification d'administrateur	50
Figure III.5: Interface ajout personnel	50
Figure III.6: Interface liste personnel.....	51
Figure III.7: Interface ajouter RDV	51
Figure III.8: Interface d'accueil (utilisateur android)	52
Figure III.9: Interface listes des personnels (coté android).....	53
Figure III.10: Interface envoyer un e-mail.....	54
Figure III.11: Interface de la liste des RDV	55

Table des illustrations

Les tables

Figure II.16: Table personnel.....	41
Figure II.17: Table de catégorie	41
Figure II.18: Table compte.....	42
Figure II.19: Table RDV	42

Introduction Générale

L'évolution des technologies de l'information et de la communication ont permis un penchant de plus en plus fort vers l'utilisation des appareils sans fils, notamment les téléphones mobiles. La généralisation de l'usage du téléphone mobile par la majeure partie de la population pousse les développeurs à proposer diverses fonctionnalités et services comme les jeux mobiles, les automatismes industriels et services basés sur la localisation.

Il est important pour tout individu de se tenir au courant de ce qui se passe autour de lui et d'avoir facilement et rapidement accès aux informations dont il a besoin quand il en a besoin. D'où vient l'idée de développer une application sous Android permettant de localiser un personnel médical dans une spécialité demandée en quelques clics.

Le but de cette application est de permettre à un patient (malade) de pouvoir solliciter un personnel médical ou paramédical pour une consultation à domicile, ou pour localiser son adresse exacte et sa disponibilité pour s'y rendre ou prendre un rendez-vous. Nous avons également pensé à garantir l'innovation, dans notre cas la prise de rendez-vous, qui éliminera l'obstacle des horaires à respecter car la prise de rendez-vous sera désormais possible à tout moment.

Ce type d'application s'avère très utile et pratique qui offre principalement deux grands avantages : Le gain significatif de temps lors de la recherche d'un personnel médical, en fournissant toutes les informations sur celui-ci afin de le rapprocher de ses patients et de le rendre plus disponible et surtout accessible en cas de besoin, et d'un autre côté, il permet la prise, le changement ou l'annulation du rendez-vous avec le personnel médical par mail, à tout moment indépendamment des horaires de travail.

Afin de mener à bien notre projet, nous l'avons structuré en trois chapitres, à savoir dans le premier chapitre «Les Systèmes d'exploitation mobile», nous allons présenter les systèmes d'exploitation conçus pour fonctionner sur un appareil mobile. Dans le deuxième chapitre « analyse et conception », nous élaborons l'analyse et la conception de notre application, pour cela on a fait recours au langage UML étant le mieux adapté pour les applications mobiles. Le troisième est le dernier chapitre « Implémentation et test », comporte la présentation de l'environnement dont lequel notre application a été réalisée, les outils

Introduction Générale

utilisés et quelques interfaces de notre application. Enfin nous terminons se mémoire par une conclusion.

Introduction générale

Sommaire

Chapitre I

Les systèmes d'exploitation mobiles



Chapitre II

Analyse et conception



Chapitre III

Implémentation et test



Conclusion générale

Bibliographie

I. Introduction

Un système d'exploitation mobile est un système d'exploitation conçu pour fonctionner sur un appareil mobile, Dans cette section, nous aborderons quelques plateformes existantes, Cela nous permettra d'avoir une idée assez générale des systèmes d'exploitation tournant sur mobiles, et d'essayer de faire le positionnement d'Android dans l'environnement des systèmes d'exploitation pour mobile.

II. Système d'exploitation des Smartphones

Les 4 principaux systèmes sont Android, iOS, Windows phone et Blackberry OS, représentant la quasi-totalité des parts de marché.

II.1. iPhone OS

iPhone OS d'Apple est le premier OS pour téléphones tactiles qui a véritablement lancé la vague des Smartphones. Apparu sur le marché en 2007, il a innové dans un domaine qui n'était pas encore développé. Cependant, iOS n'a pas beaucoup évolué depuis sa création. iOS se caractérise par une interface peu chargée, qui permet l'exécution de tâches rapidement pour les utilisateurs réguliers, mais qui laisse peu de place à la personnalisation. Dans de nombreux cas on observe le positionnement de mêmes éléments qui diffère d'une application à l'autre. Une étude a montré que les utilisateurs sont fréquemment ralentis par ce manque d'homogénéité, ce qui entraîne un grand nombre d'erreurs [1].



Figure I.1: Icône représentant le système iOS.

II.2. BlackBerry

Tout comme l'iPhone, le BlackBerry est aussi un téléphone très Utilisé.

La fonction majeure qui a fait décoller le BlackBerry était le push mail. L'utilisateur n'a alors plus besoin de consulter périodiquement sa boîte pour vérifier s'il n'a pas de nouveaux messages. Ceux-ci lui parviennent directement comme un simple SMS. Cette fonctionnalité est assurée par les serveurs d'infrastructure du fabricant RIM (Research In Motion) avec un protocole propriétaire. Le mail est donc le point fort des BlackBerry qui a fait son succès auprès des cadres et dirigeants [2].



Figure I.2: Exemple d'un téléphone utilisant BlackBerry OS.

II.3. Symbian

Symbian est très populaire (particulièrement en Europe), c'est le leader mondial du système d'exploitation pour Smartphones. Il offre une plateforme flexible, ce qui veut dire que les constructeurs de téléphones mobiles peuvent facilement y ajouter leurs technologies et infrastructures. De plus, il est soutenu par les grandes manufactures de l'industrie mobile comme Sony Ericsson, Motorola, et Nokia. Ce dernier étant constructeur numéro un des mobiles dans le monde et principal actionnaire de Symbian, cela garantit la position du Symbian dans le marché des systèmes d'exploitation pour les Smartphones. Néanmoins depuis quelques années, la part de marché de Symbian diminue à cause de concurrence avec d'autres plateformes [3].



Figure I.3: Exemple d'un téléphone sous OS Symbian.

II.4. Windows Mobile

Windows Phone, apparu en 2010, a su s'inspirer de ses concurrents et de son expérience passée dans le domaine du mobile. Proposant une interface simple et épurée ainsi que de multiples possibilités de personnalisation, il permet d'arriver rapidement à l'exécution d'une tâche ce qui a particulièrement séduit les utilisateurs. C'est d'ailleurs Windows Phone qui se trouve être l'OS mobile le plus adapté aux utilisateurs [4].

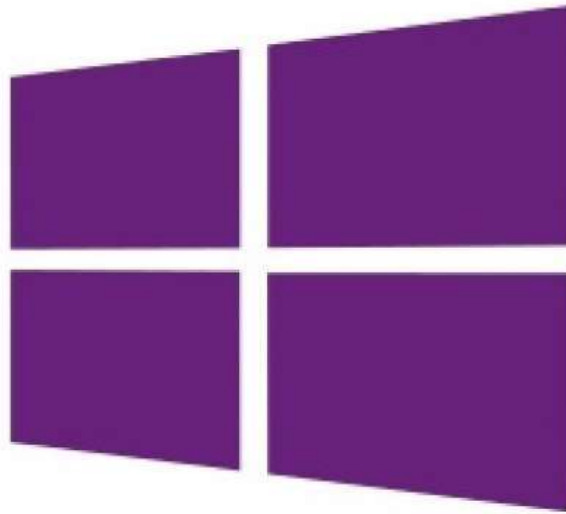


Figure I.4: Icône représentant le système Windows Mobile.

II.5. Android OS

Android est un OS basé sur linux conçu par Google pour les appareils mobiles. Il est gratuit et complètement ouvert C'est-à-dire que le code source et les APIs sont ouvertes, il a été développé par l'Open Handset Alliance en 2007 et il est devenu une plateforme ouverte en 2008. Android a intégré plusieurs services de Google pour accéder rapidement aux services d'internet comme Gmail, YouTube, Google Talk, Google Calendar et Google Maps [5].

III. Comparaison entre les différents Systèmes d'exploitation mobiles

Dans ce qui suit, nous donnons une comparaison entre les différents systèmes d'exploitation mobiles au niveau de leurs performances en premier lieu, puis de leur popularité et nous examinons les parts du marché de chacun.

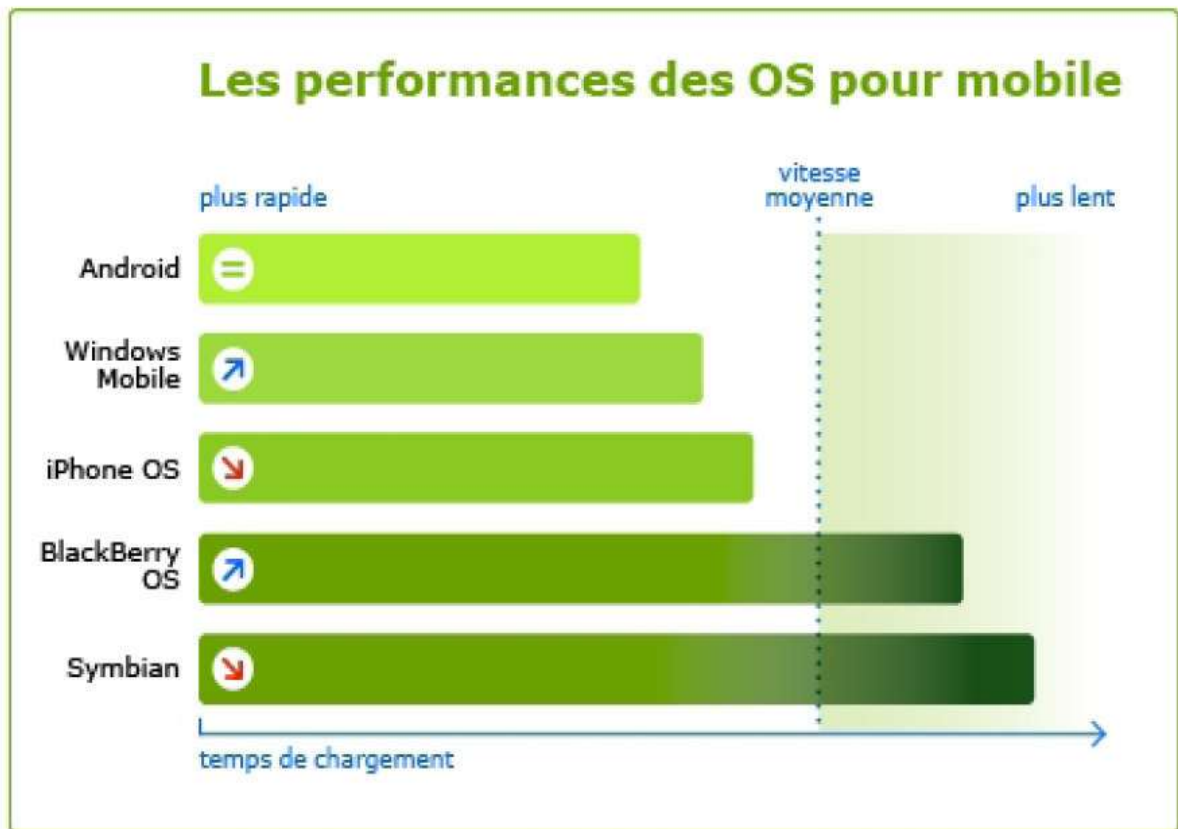


Figure I.5: Performances des Systèmes d'Exploitation mobiles.

L'histogramme montre les différents systèmes d'exploitation mobiles comparés à base de leurs performances par temps de chargement, on remarque que le système Android présente le meilleur taux des performances d'un OS mobile par rapport aux autres systèmes. Ces performances expliquent les différentes parts acquises dans le marché de l'OS mobile.

IV. Présentation de la plateforme Android

Cette partie présente Android sous un autre angle que celui de l'utilisateur, qui est celui d'une vue de l'intérieur, en explorant les aspects techniques internes.

IV.1. Architecture d'Android

La plate-forme Android est composée de différentes couches :

IV.1.1. Application

Le système Android peut exécuter un grand nombre d'applications incluant les clients de messagerie, les programmes de gestion d'SMS, les gestionnaires de rendez-vous, les navigateurs, ...etc. Toutes ces applications sont écrites en Java.

IV.1.2. Framework Applicatif

C'est un ensemble de fonctions permettant aux applications d'exploiter au maximum les ressources de l'équipement, de communiquer avec le système et les autres applications (Accès à divers informations, exécution en tâche de fond, afficher des notifications dans la barre d'état, déclencher des alarmes, ...etc.). Ces fonctions sont les mêmes que celles utilisées par le système lui-même. Ce qui donne aux applications toute la puissance nécessaire.

IV.1.3. Les Librairies

C'est un ensemble de bibliothèques développés en C/C++. Ces librairies sont utilisées par divers composant d'Android dont le Framework applicatif.

Les composants majeurs de la plate-forme Android sont résumés sur le schéma suivant :

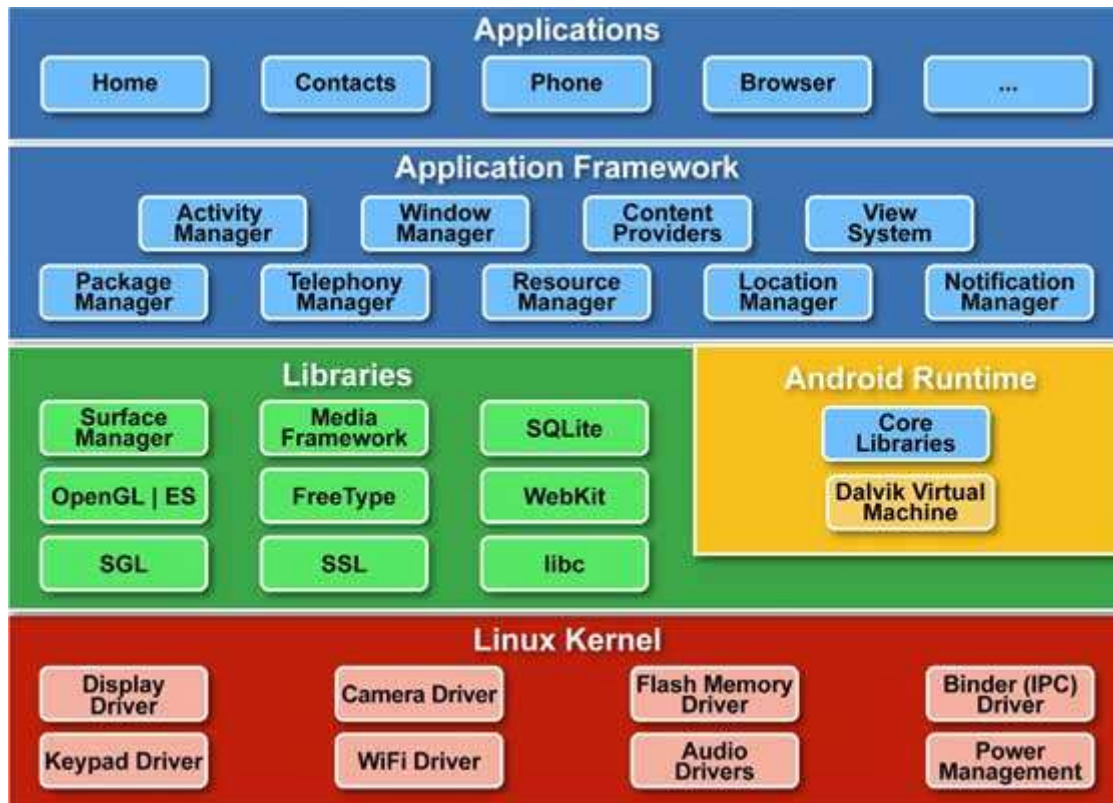


Figure I.6: Architecture d'un système Android.

IV.1.4. Environnement d'exécution Android

L'environnement d'exécution d'Android est la machine virtuelle Dalvik avec ses propres bibliothèques adaptées aux équipements mobiles. Dans la machine virtuelle Dalvik, les programmes sont écrits en Java et compilés avec les outils de java pour obtenir un byte code qu'on recompile encore avec un outil spécifique (dex) pour obtenir un code adapté à la machine Dalvik.

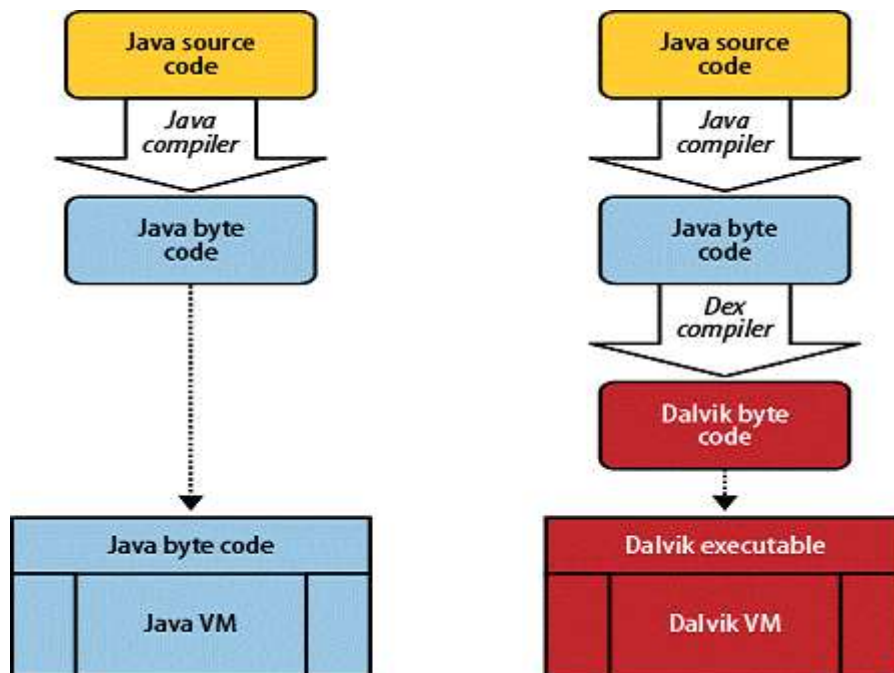


Figure I.7 : Différence entre la JVM et Dalvik.

IV.1.5. Le noyau linux

Android repose sur le noyau Linux version 2.6 pour la gestion de la mémoire, des processus et les pilotes des différents périphériques.

IV.2. Les concepts d'une application Android

Une application android est constituée d'un certain nombre de composants (Activity, service, Content Provider et Broadcast Receiver)

IV.2.1. L'activité (Activity)

Une activité est un composant fournissant un écran avec lequel les utilisateurs interagissent avec l'application. Quand on exécute n'importe quelle application Android on remarque qu'une première fenêtre s'ouvre à nous avec laquelle il est possible d'interagir (introduire ou extraire des données), elle possède donc une interface graphique. Dans certains cas, en appuyant sur un bouton par exemple ou un TextView cela nous mène vers une autre activité, et ainsi de suite.

Cela est possible grâce au mécanisme d'Intent. Une application donc est un assemblage d'activités entre lesquelles il est possible de naviguer.

Une activité remplit tout l'écran, une application ne peut en afficher qu'une seule à la fois. De plus, elle contient des informations sur l'état actuel de l'application. Ces informations s'appellent le Context. Toute activité peut passer à tout moment en arrière-plan dans une pile d'attente lorsque son exécution est interrompue comme lors d'un appel entrant [6].

IV.2.1.1.Etat d'une activité

Une activité peut se trouver dans trois états qui se différencient surtout par leur visibilité :

Active (Resumed)

L'activité est visible en totalité. Elle est sur le dessus de la pile, c'est elle qui a le focus. C'est ce que l'utilisateur consulte et peut l'utiliser dans son intégralité et agir directement dessus.

Suspendue (Posed)

L'activité est partiellement visible à l'écran. C'est le cas lors de la réception d'un SMS et qu'une fenêtre semi-transparente se pose devant l'activité pour afficher le contenu du message. Ce n'est pas sur l'activité suspendue qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application sous-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser de l'application qui l'obstrue, puis il pourra à nouveau interagir avec elle.

Arrêtée (Stopped)

L'activité est tout simplement masquée par une autre activité, on ne peut plus la voir. L'application n'a évidemment plus le focus, et on ne peut pas agir dessus. Le système retient son état pour pouvoir reprendre, mais il peut arriver que le système tue l'application pour libérer de la mémoire système.

Les transitions d'états d'une activité sont captées par les méthodes suivantes :

➤ **onCreate ()**

Est appelée au début de la création de l'activité et n'est appelée qu'une seule fois. Elle joue le rôle du constructeur en permettant d'initialiser des variables, affecter des listener...

➤ **onRestart ()**

Appelée après un nouveau démarrage de la même activité (quand l'activité était arrêtée).

➤ **onStart ()**

L'activité va devenir visible. Cette méthode sert à lancer les animations, ou généralement tout ce qui est liée à l'affichage graphique, car elle est également appelée lors d'un retour de focus sur l'activité (dans ce cas onRestart est appelé avant).

➤ **onResume ()**

L'activité est maintenant visible, Cette méthode sera exécutée lorsque l'activité résume son exécution après la suspension (pause) et que l'activité commence à interagir avec l'utilisateur.

➤ **onPause ()**

Méthode qui sert à arrêter une activité temporairement.

➤ **onStop ()**

L'activité ne sera plus visible, cachée par une autre activité qui est en premier plan. Une activité stoppée est aussi en vie, elle est encore en mémoire mais elle n'est pas rattachée au gestionnaire des fenêtres du système Android. Elle peut être tuée par le système Android en cas de besoin en mémoires.

➤ **onDestroy()**

L'activité va être détruite. La destruction opère quand quelqu'un appelle cette méthode ou quand c'est le système qui décide de tuer l'activité pour économiser de l'espace.

IV.2.1.2. Cycle de vie d'une activité

Une activité n'a pas de contrôle direct sur son propre état, il s'agit plutôt d'un cycle rythmé par les interactions avec le système et d'autres applications. Voici un schéma qui représente ce que l'on appelle le cycle de vie d'une activité, c'est-à-dire qu'il indique les étapes que va traverser notre activité pendant sa vie, de sa naissance à sa mort.

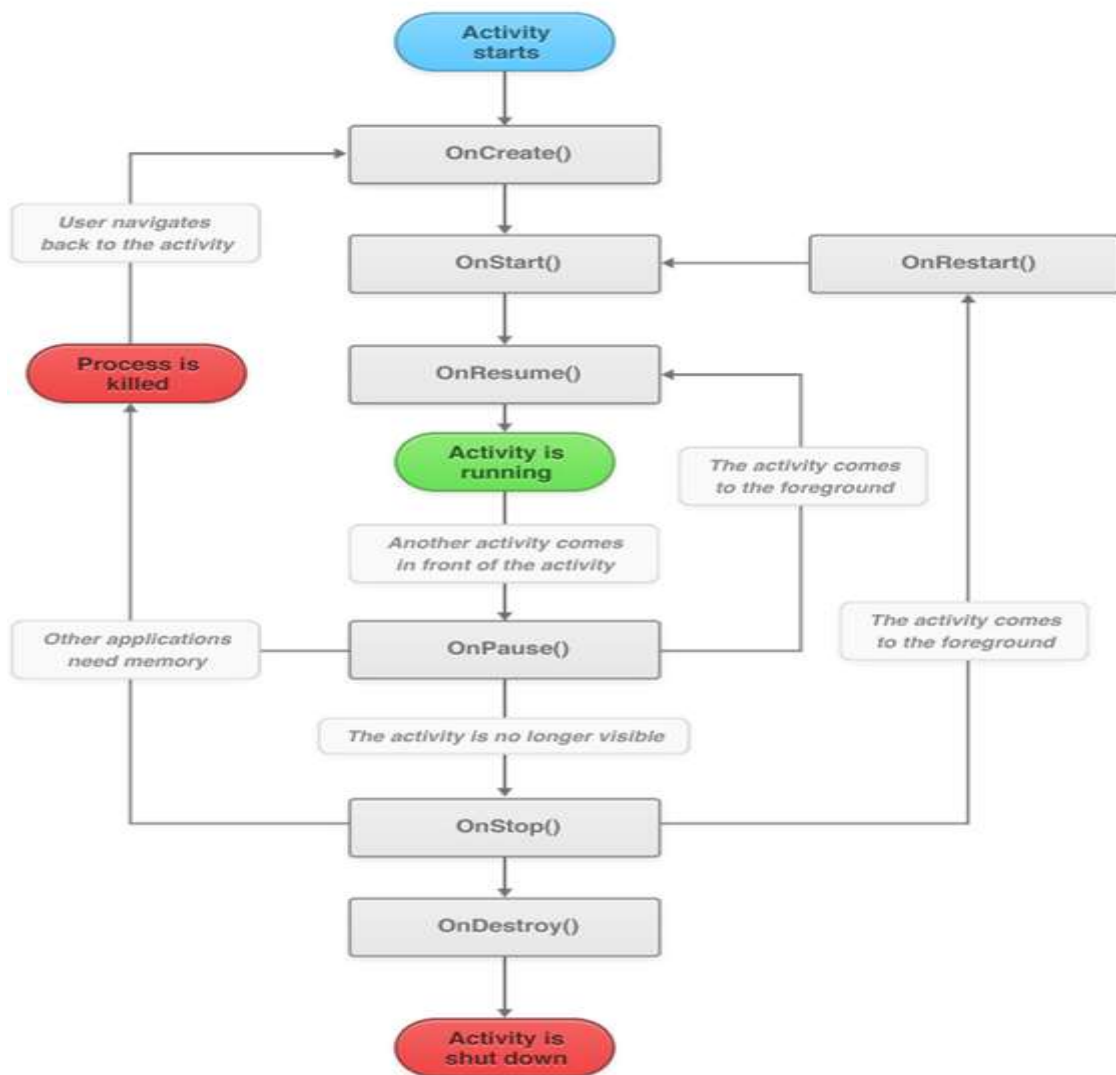


Figure I.8: Cycle de vie d'une activité.

IV.2.2. Les Services

Les services sont des tâches qui peuvent être lancées avec ou sans intervention de l'utilisateur. Elles s'exécutent dans le background de l'application et peuvent se terminer soit après la finalisation de la tâche, soit à travers une intervention externe. Les services représentent également une fonctionnalité d'une application exposée à d'autres applications. Il est important de mentionner que le service ne fournit pas d'interface graphique (UserInterface).

Notre Player audio (lecteur de musique), par exemple, permet d'écouter la musique tout en consultant nos emails, etc.... Cette fonctionnalité n'est possible qu'à l'aide des Services [7].

IV.2.3. Content Provider

Les fournisseurs de contenu offrent un moyen de partager des données entre les applications qui peuvent être stockées sur le système de fichier local, sur une base de données SQLite ou le réseau (FTP, Web, etc.). Les autres applications peuvent accéder à ces données (ou les modifier si l'application le permet) en utilisant un content provider.

Un content provider est une classe qui implémente un ensemble de méthodes standards permettant à d'autres applications de consulter et modifier les données prises en charge par ce content provider. Par exemple, le système Android fournit un content provider fournissant la liste des contacts téléphoniques qu'une autre application peut utiliser [8].

IV.2.4. Broadcast Receiver

Les broadcasters sont les diffuseurs d'évènements/messages via des intentions (intentions dites broadcast pour les différencier des intentions plus classiques). Les messages ainsi diffusés pourront être réceptionnés par plusieurs applications, les applications qui se seront abonnées à ces broadcasts (diffusions).

IV.2.5. Intent

Trois composants d'application sont lancés par les Intents : Les activités, Les services, Les receveurs de diffusion. Les Intents sont l'une des pierres angulaires de la plateforme Android. Nous pouvons les comparer à des actions ou même à des intentions, ils permettent de dialoguer à travers le système à partir de canaux qui leur sont dédiés. Quand le mobile reçoit un appel, la plateforme lance un Intent signalant l'arrivée d'un appel, de même pour un SMS. Nous les utiliserons dans notre application pour dialoguer à l'intérieur de celle-ci pour :

- ✓ Naviguer entre les activités.
- ✓ Surveiller les clicks sur l'AppWidget.
- ✓ Lancer le service de téléchargement des données météorologiques.

V. Historique et versions d'Android

V.1. Open Handset Alliance

Afin de promouvoir ce nouveau système d'exploitation ouvert, Google a su fédérer autour de lui un consortium d'une trentaine d'entreprises. L'Open Handset Alliance (abrégé OHA) est un consortium regroupant de grands constructeurs et développeurs de logiciels dans le but de développer des normes ouvertes pour les appareils de téléphonie mobile.

Ce consortium a été créé le 5 novembre 2007. Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6 [9].

V.2. Bugdroïde

Pour représenter Android la famille Google a utilisé un personnage nommé Bugdroid qui est un petit robot vert. Ce personnage est sous licence « creative commons by (3.0) » et peut donc être utilisé librement. Le site Engadget annonce que Bugdroid, le logo d'Android, serait en fait un personnage d'un jeu des années 1990 sur Atari Gauntlet, TheThird Encounter [10].



V.3. Versions d'Android

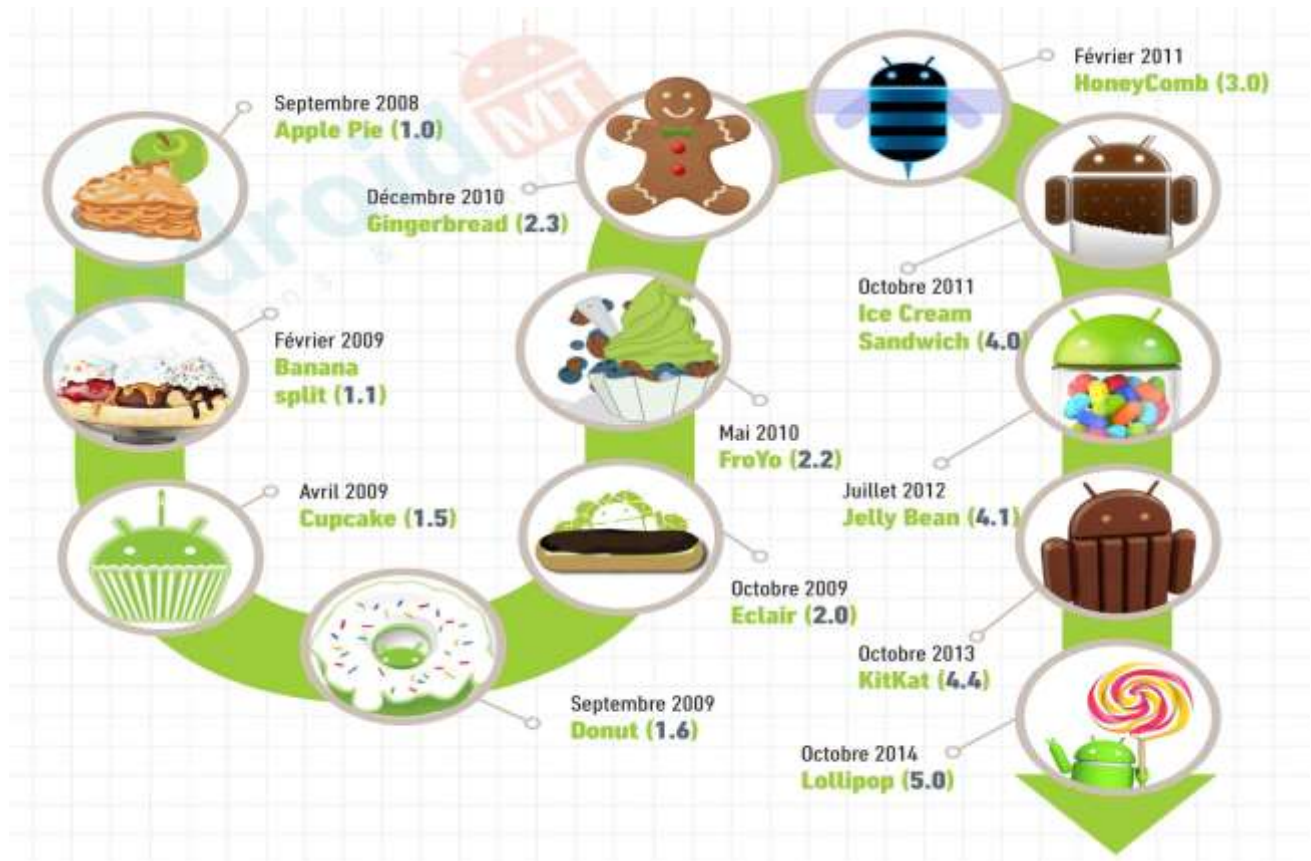


Figure I.9: Versions d'Android.

VI. Différents équipements utilisant Android

Le succès d'Android est indéniable, les fournisseurs l'ont bien compris, d'où l'apparition sur le marché de plus en plus d'appareils utilisant ce système d'exploitation :

VI.1. Les Smartphones

Est un téléphone intelligent, disposant d'un écran tactile, fonctionnant sur des réseaux haut débit et offrant de nombreuses fonctionnalités, dont la consultation du courrier électronique

et localisation par GPS. Le premier mobile commercialisé sous Android est le HTC Dream/G1 produit par la firme Taïwanaise HTC, lancé aux États-Unis le 22 octobre 2008. Depuis le lancement en 2007 de l'iPhone d'Apple, la progression de leurs ventes est spectaculaire et elles ont dépassé en 2012 celles des téléphones mobiles classiques.

VI.2. Les Tablettes

En septembre 2010, plusieurs marques comme Samsung ont utilisé le système Android pour les tablettes (version 2.2 Fro Yo) Android a été ainsi porté sur d'autres appareils comme la HP TouchPad, le portage a été réalisé début 2012 basé sur la version Ice Cream Sandwich, et fin 2011 avec la version 7 de CyanogenMod basé sur Gingerbread.

VI.3. Les télévisions

Le 5 avril 2010, la première télévision sous Android est dévoilée. Celle-ci est développée par l'entreprise suédoise People of Lava et se nomme Scandinavia. Elle possède les différentes applications connectant aux réseaux sociaux (facebook.....), elle possède un navigateur Web ainsi qu'un client de messagerie électronique.

VI.4. Consoles de jeux vidéo

Une console de jeux vidéo portable sous Android portant le nom GamePad est commercialisée depuis septembre 2012.

VII. Conclusion

Ce chapitre nous a permis d'aborder les différents systèmes d'exploitation conçus pour fonctionner sur les appareils mobiles. Ainsi on a étudié le positionnement d'Android par rapport aux autres systèmes en spécifiant ses notions de bases, ses caractéristiques et ses possibilités d'utilisation.

La popularité d'Android et sa généralisation sur la plupart des équipements du public, smart, table, téléphone nous ont poussé à choisir ce système comme cible pour notre application.

I. Introduction

Pour une meilleure compréhension du travail de réalisation d'un projet informatique, les étapes d'analyse et de conception sont vraisemblablement indispensables pour la présentation globale de ce projet. Pour modéliser ces deux étapes nous avons utilisé le langage de modélisation UML (Unified Modeling Language).

Dans ce chapitre nous allons voir la conception de notre application en commençant par la problématique et la solution proposée ensuite nous passons à la présentation de l'UML : sa définition ses diagrammes ainsi que la démarche de modélisation adoptée. Puis, nous entamons l'analyse par la présentation des diagrammes de contexte et de cas d'utilisation. Nous concluons par la conception en présentant les diagrammes de séquence détaillés pour quelques cas d'utilisation et le diagramme d'activité.

II. La problématique et la solution proposée

Tous les jours, les gens, en particulier les patients, sont amenés à chercher à joindre un ou plusieurs personnels d'une structure médicale ou paramédicale. Vu la grandeur de la ville, et les différents emplacements, il leur est difficile de s'auto-orienter vers le personnel médical adéquat.

Pour cela, nous avons développé pour notre projet de fin d'étude, une application mobile qui permettra aux utilisateurs d'avoir la possibilité de localiser un médecin (généraliste ou spécialiste, une infirmière, un dentiste, etc.) dans une ville. Ainsi pouvoir le contacter par mail ou par appel téléphonique, pour s'y rendre et même pour prendre un rendez-vous. D'autre part, nous avons mis en place une application Web qui permet à l'administrateur de gérer facilement toutes les tables de la base de données ainsi une page web pour les médecins leurs permettant la mise à jour de leurs disponibilités.

Le sujet est certes intéressant car l'application dont nous parlons peut avoir une vraie valeur ajoutée dans la vie de chacun d'entre nous.

III. Présentation d'UML




III.1. Définition

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. En effet UML est un langage avec une syntaxe et des règles bien définies qui tentent à réaliser les buts décrits grâce à une représentation graphique formée de diagrammes et une modélisation textuelle qui vient enrichir la représentation graphique.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.4.1 depuis aout 2011.

III.2. Les diagrammes UML

Les diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Ils sont au nombre de quatorze, répartis en trois catégories :

-  Diagrammes structurels ou statiques.
-  Diagrammes comportementaux.
-  Diagrammes d'interaction ou dynamiques.

III.3. La démarche de modélisation

Une démarche de modélisation est la façon d'agencer, d'utiliser et de formaliser les diagrammes. Son objectif est de parvenir à modéliser le cycle de vie d'un logiciel. Dans notre démarche de modélisation, nous allons formaliser deux phases du cycle de vie :

- Première phase : c'est la phase d'analyse et pour la modéliser nous utiliserons le diagramme de contexte et le diagramme de cas d'utilisation.
- Deuxième phase : c'est la phase de conception, nous utiliserons le diagramme de séquence détaillé, le diagramme de d'activités ainsi le diagramme de classe globale.

IV. Analyse

Correspondant à la phase qui répond à la question « que fait le système », l'analyse est l'une des étapes les plus importantes de la modélisation. Elle permet de modéliser le domaine d'application, d'analyser l'existant et les contraintes de réalisation.

Cette partie comprend l'identification des besoins fonctionnels du système, des acteurs et leurs interactions avec le système avant de conclure sur les cas d'utilisation.

IV.1. Identification des acteurs

Un acteur représente un ensemble de rôles joués par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Dans notre cas, nous avons choisi de nous adresser au grand public pour offrir nos services, nous dégageons alors un seul acteur qui va interagir avec l'application Android qui est :

L'utilisateur de l'application Android

Cet acteur a pour responsabilité de lancer l'application, puis accéder au menu des tâches et notre application et delà, entre autre, localiser tous le personnel médical c.à.d. il a comme tâches :

- ✓ Chercher personnel médical ou paramédicale.
- ✓ Afficher profil du personnel.
- ✓ Prendre un rendez-vous.
- ✓ Appeler le personnel.
- ✓ Envoyer un mail.

Remarque : l'utilisateur est le patient (malade)

D'autre part pour assurer la gestion de la partie mobile, nous avons mis en place une application Web qui à son tour aura deux acteurs qui sont :

Un administrateur

C'est une personne qui s'occupe de la mise à jour de la base de données, il effectue les tâches suivantes :

- ✓ S'authentifier par un identifiant et un mot de passe.
- ✓ Gérer personnel médical ou paramédical.

Personnel médical ou paramédical

Tout personnel médical ou paramédical disposant d'un compte peuvent gérer leurs disponibilité, c.à.d. ils ont comme tâches :

- ✓ S'authentifier par un identifiant et un mot de passe.
- ✓ Gérer RDV.

IV.2. Diagramme de contexte

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et l'environnement extérieur, notre diagramme de contexte est donné par la figure suivante :

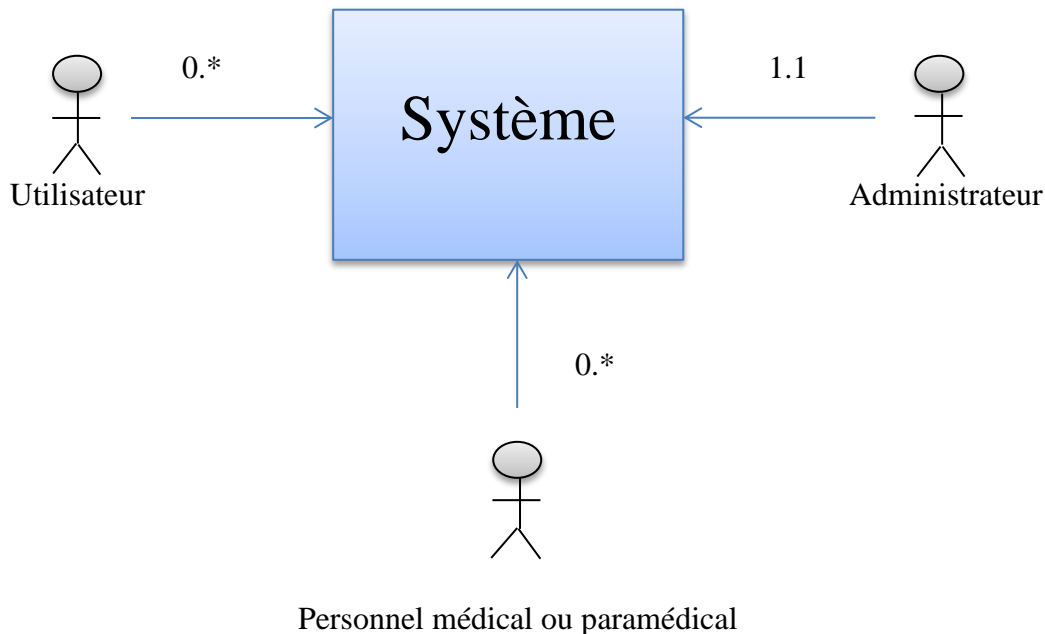


Figure II.1. Diagramme de contexte.

IV.3. Les cas d'utilisation

Un cas d'utilisation représente un ensemble de séquence d'actions qui sont réalisées par les systèmes et qui produit un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le système devra faire, sans spécifier comment le faire.

Il existe trois types de relations standards entre cas d'utilisations qui sont proposés par UML :

- « include » le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié.
- « extend » le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié.
- « Généralisation » : les cas d'utilisations descendant héritent des propriétés de leurs parents.

Dans notre cas nous distinguons les diagrammes des cas d'utilisations suivants :

1. Diagramme de cas d'utilisation pour l'application mobile (coté client)

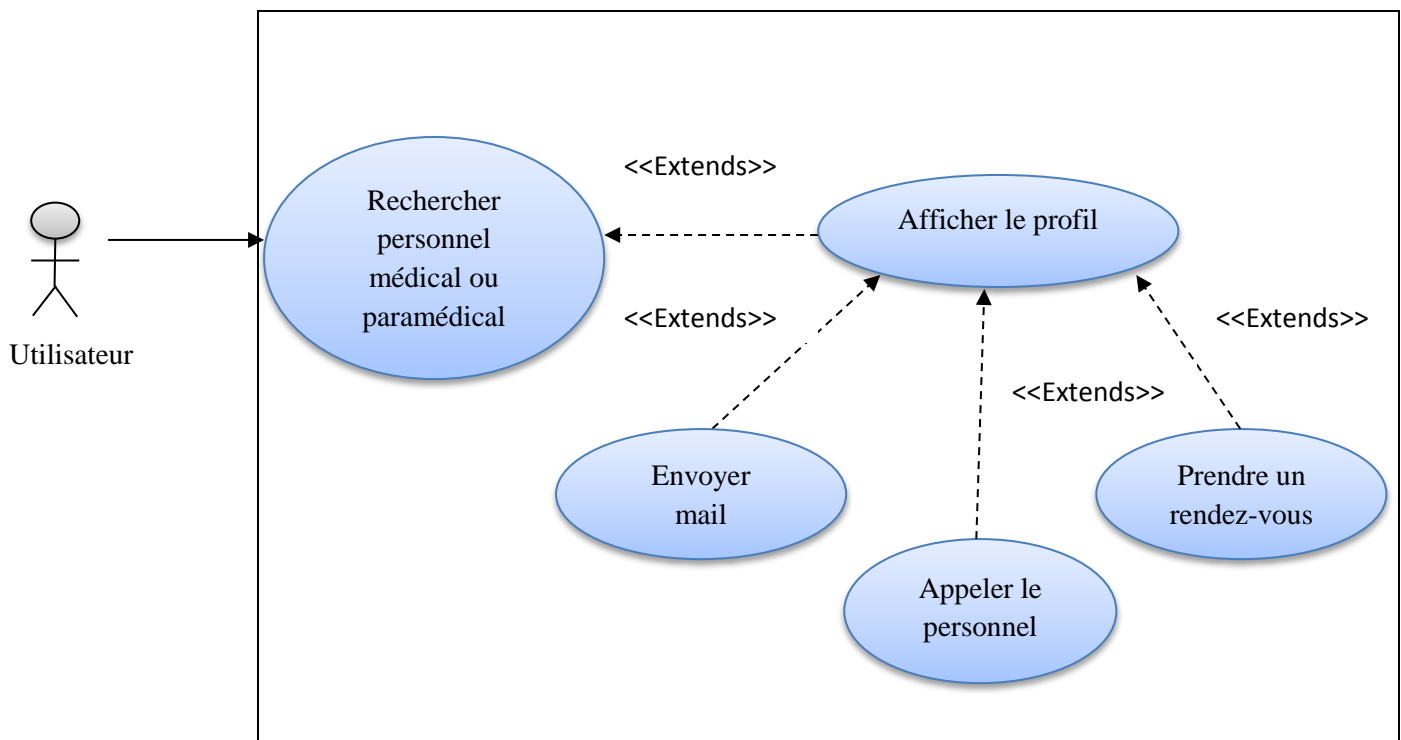
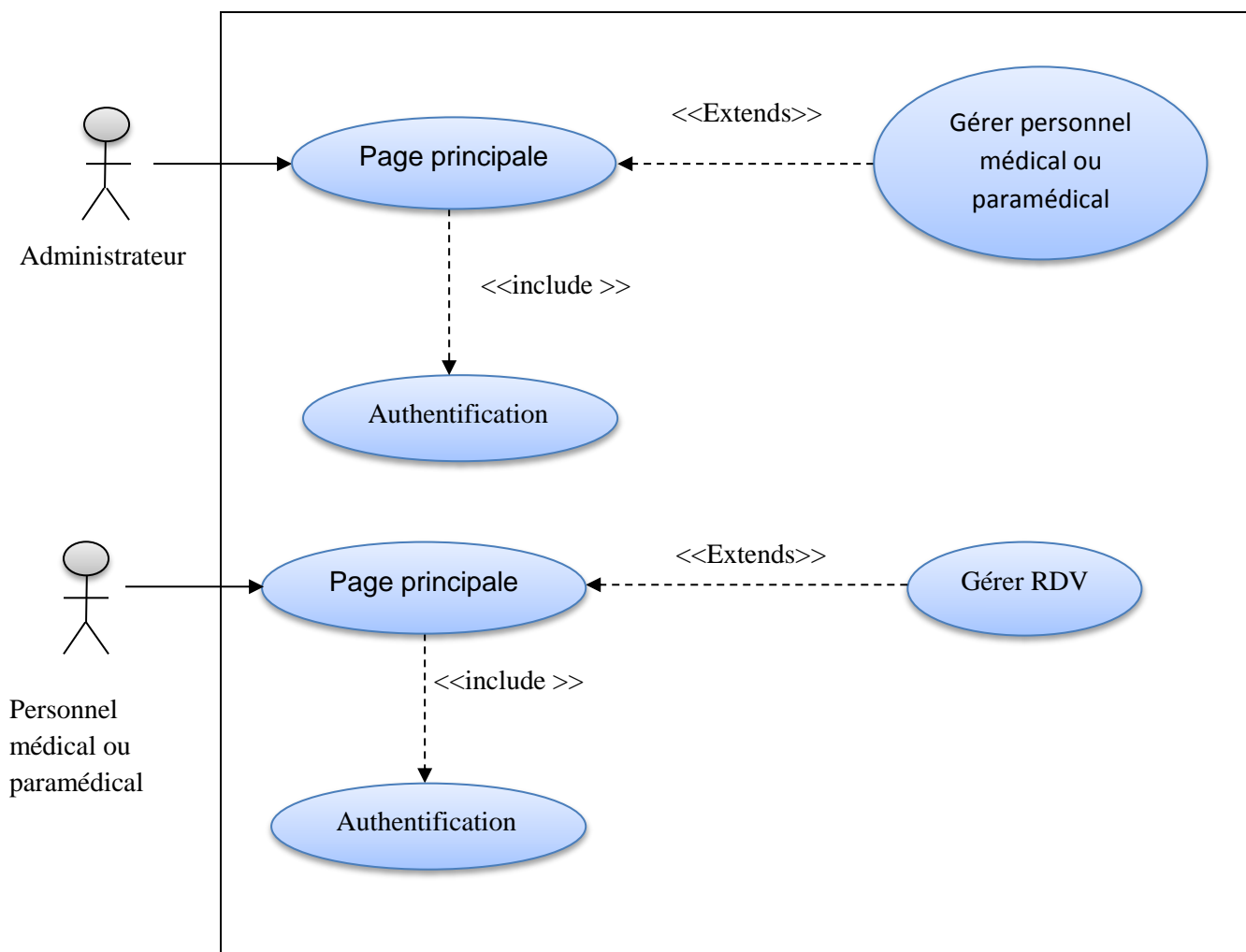


Figure II.2. Diagramme de cas d'utilisation pour l'application mobile.

Selon ce diagramme, l'utilisateur de notre application peut rechercher un médecin sans authentification préalable, nous avons fait ce choix afin d'alléger l'utilisation de notre application et de la rendre la plus disponible et efficace possible.

2. Diagramme de cas d'utilisation pour la partie Web :

**Figure II.3.** Diagramme de cas d'utilisation pour la partie web.**IV.4. Description des cas d'utilisation avec des scénarios**

Pour détailler le déroulement d'un cas d'utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système.

Dans ce qui suit nous décrivons donc quelques cas d'utilisation de notre système.

IV.4.1. Description du cas d'utilisation « Rechercher personnel médical ou paramédical »

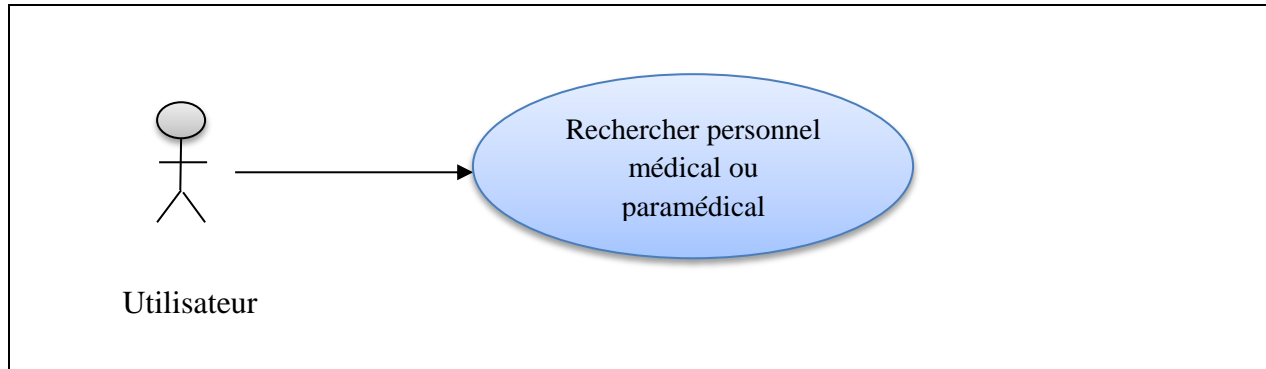


Figure II.4. Diagramme de cas d'utilisation « Rechercher personnel médical ou paramédical ».

Acteur	Utilisateur
Pré-condition	L'application mobile est lancée avec succès.
Post-condition	Liste de personnel médical ou paramédical, correspondant aux critères de l'utilisateur, affichée avec succès.
Description de scénario de base	<ol style="list-style-type: none"> 1. L'utilisateur atteint la page principale. 2. Le système affiche la page de choix de critère de recherche. 3. L'utilisateur sélectionne un critère. 4. L'utilisateur saisi la spécialité qui veut. 5. Le système affiche la liste de tous les médecins correspondant aux critères choisis.

IV.4.2. Description du cas d'utilisation « Afficher profil »

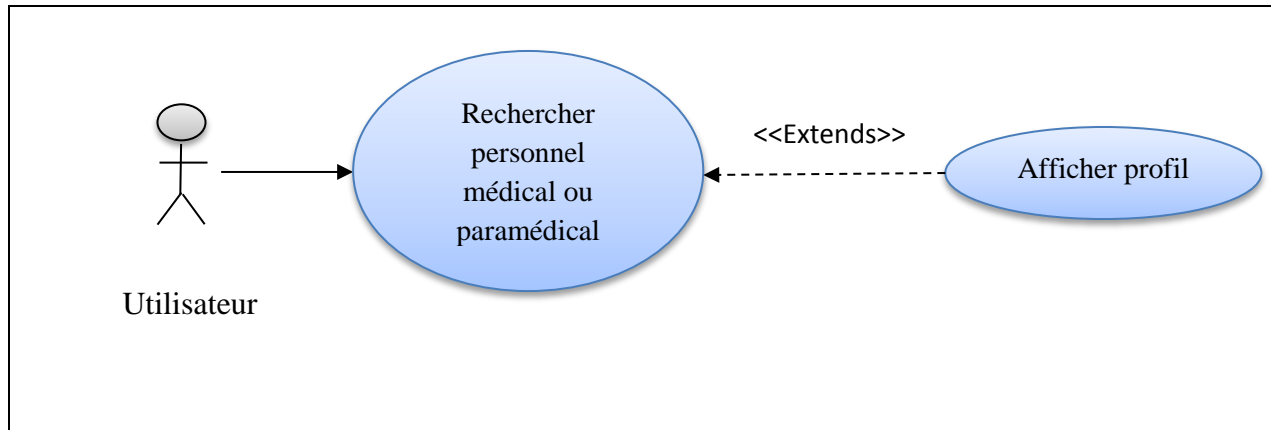
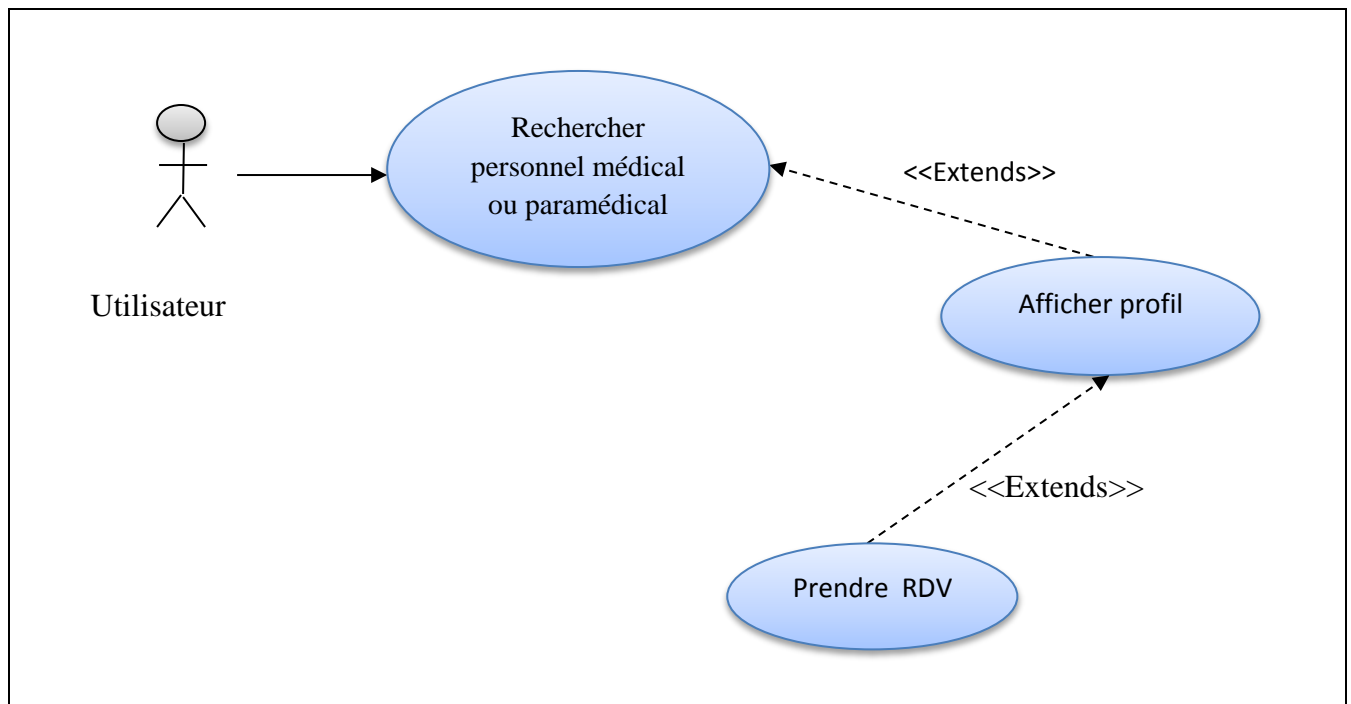


Figure II.5. Diagramme de cas d'utilisation « Afficher profil ».

Acteur	Utilisateur
Pré-condition	Liste de personnel recherchée affichée avec succès.
Post-condition	Affichage du profil du personnel médical ou paramédical.
Description de scénario de base	<ol style="list-style-type: none"> 1. L'utilisateur clique sur un personnel. 2. Le système affiche le profil (nom, prénom, e-mail, adresse, numéro de téléphone, sa disponibilité...).

IV.4.3. Description du cas d'utilisation « Prendre RDV »

**Figure II.6.** Diagramme de cas d'utilisation « Prendre RDV ».

Acteur	Utilisateur
Pré-condition	Profil du personnel médical recherché affichée avec succès.
Post-condition	Le rendez-vous est pris avec succès.
Description de scénario de base	<ol style="list-style-type: none"> 1. L'utilisateur atteint le profil du personnel. 2. L'utilisateur clique sur voir RDV. 3. Le système affiche la liste des RDVs disponible. 4. L'utilisateur clique le RDV qui le convient. 5. Le système affiche rendez-vous pris.

IV.4.4. Description du cas d'utilisation « Appeler personnel médical ou paramédical »

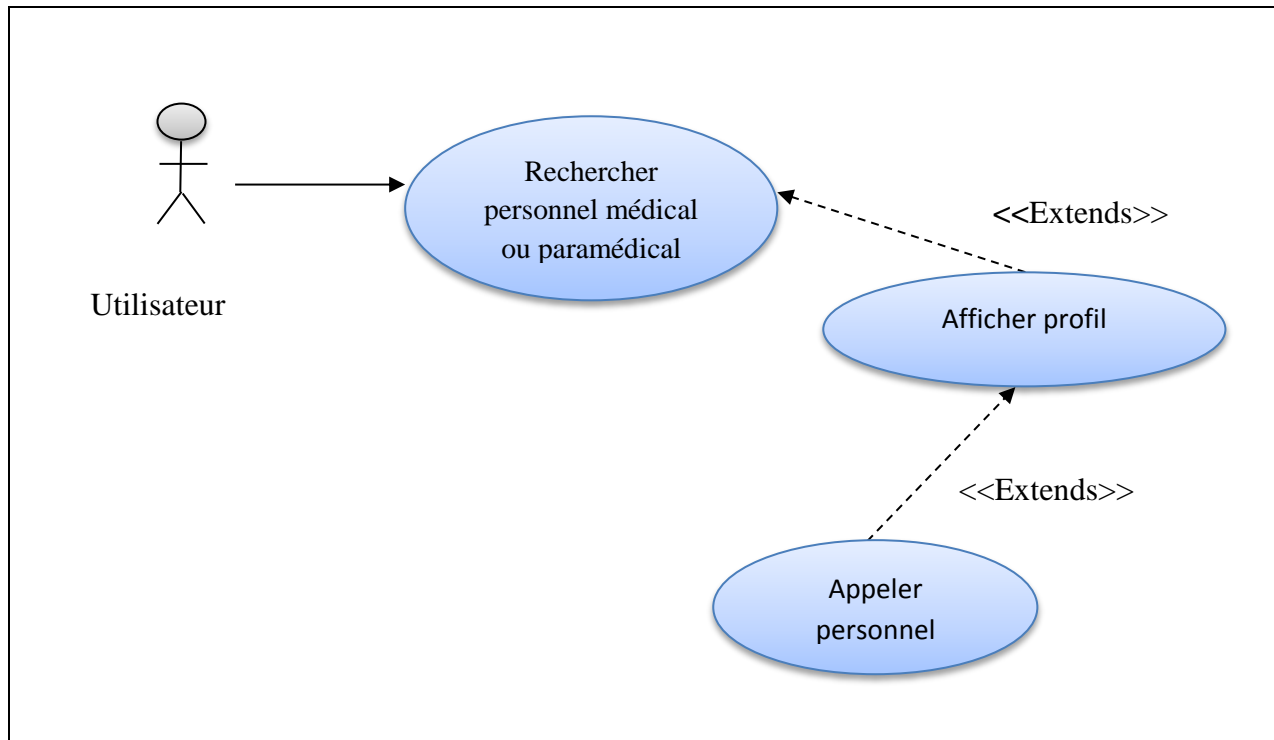


Figure II.7. Diagramme de cas d'utilisation « Appeler personnel médical ou paramédical ».

Acteur	Utilisateur
Pré-condition	Profil du personnel recherché affichée avec succès.
Post-condition	L'application lance un appel téléphonique au personnel choisis.
Description de scénario de base	<ol style="list-style-type: none"> 1. L'utilisateur atteint le profil du personnel choisis. 2. L'utilisateur clique sur le numéro de téléphone du personnel. 3. le système propose passe l'appel depuis la carte SIM.

IV.4.5. Description du cas d'utilisation «Authentification Administrateur »

Acteur	Administrateur
Pré-condition	Avoir login et mot de passe correct.
Post-condition	Administrateur authentifié.
Description de scénario de base	<ol style="list-style-type: none">1. Atteindre la page «Authentification »2. Saisit les informations d'identification (Pseudo, mot de passe)3. Cliquer sur «connecter ».4. Le système affiche un message d'erreur d'identification si les informations saisies sont incorrectes ou si elles ne correspondent pas à l'administrateur.5. Affichage de la page principal de l'espace administrateur.

IV.4.6. Description du cas d'utilisation « Ajouter personnel médical ou paramédical coté administrateur»

Acteur	Administrateur
Pré-condition	Administrateur authentifié.
Post-condition	Personnel ajouté avec succès.
Description de scénario de base	<ol style="list-style-type: none">1. Cliquer sur le bouton «personnel».2. Cliquer sur le bouton «Ajouter un personnel».3. Remplir le formulaire d'ajout et valider.4. Le system affiche une erreur si l'un des champs obligatoire n'est pas rempli.5. Sinon une page de confirmation est affichée suivie de l'identifiant, nom d'utilisation et un mot de passe concernant le personnel médical qui vient d'être ajouté.

IV.4.7. Description du cas d'utilisation «Ajouter RDV coté personnel médical ou paramédical »

Acteur	Personnel médical ou paramédical.
Pré-condition	RDV ajouté avec succès.
Post-condition	Personnel médical authentifié.
Description de scénario de base	<ol style="list-style-type: none">1. Cliquer sur le bouton RDV.2. Cliquer sur ajouter RDV.3. Remplir le formulaire d'ajout.4. Cliquer sur ajouter.5. Le système affiche RDV ajouté avec succès.

IV.5. Conception

Le processus de conception de notre application repose sur l'organisation conceptuelle, logique et physique des données collectées durant la phase analyse. En effet, elle s'appuie essentiellement sur quelques diagrammes du langage de modélisation UML.

IV.5.1. Les diagrammes de séquence

Le diagramme de séquence est un diagramme d'objet qui collabore les messages qui s'envoient les uns aux autres présentés de façon chronologique et montrant comment les cas d'utilisation sont réalisés. Le diagramme de séquence est de plus en plus utilisé, il permet de décrire la dynamique d'un système et faire le lien entre les diagrammes de cas d'utilisation et le diagramme de classe.

Dans notre cas on s'intéresse à effectuer la représentation des diagrammes de séquence pour quelques cas d'utilisation.

IV.5.1.1. Diagramme de séquence « Rechercher personnel médical ou paramédical »

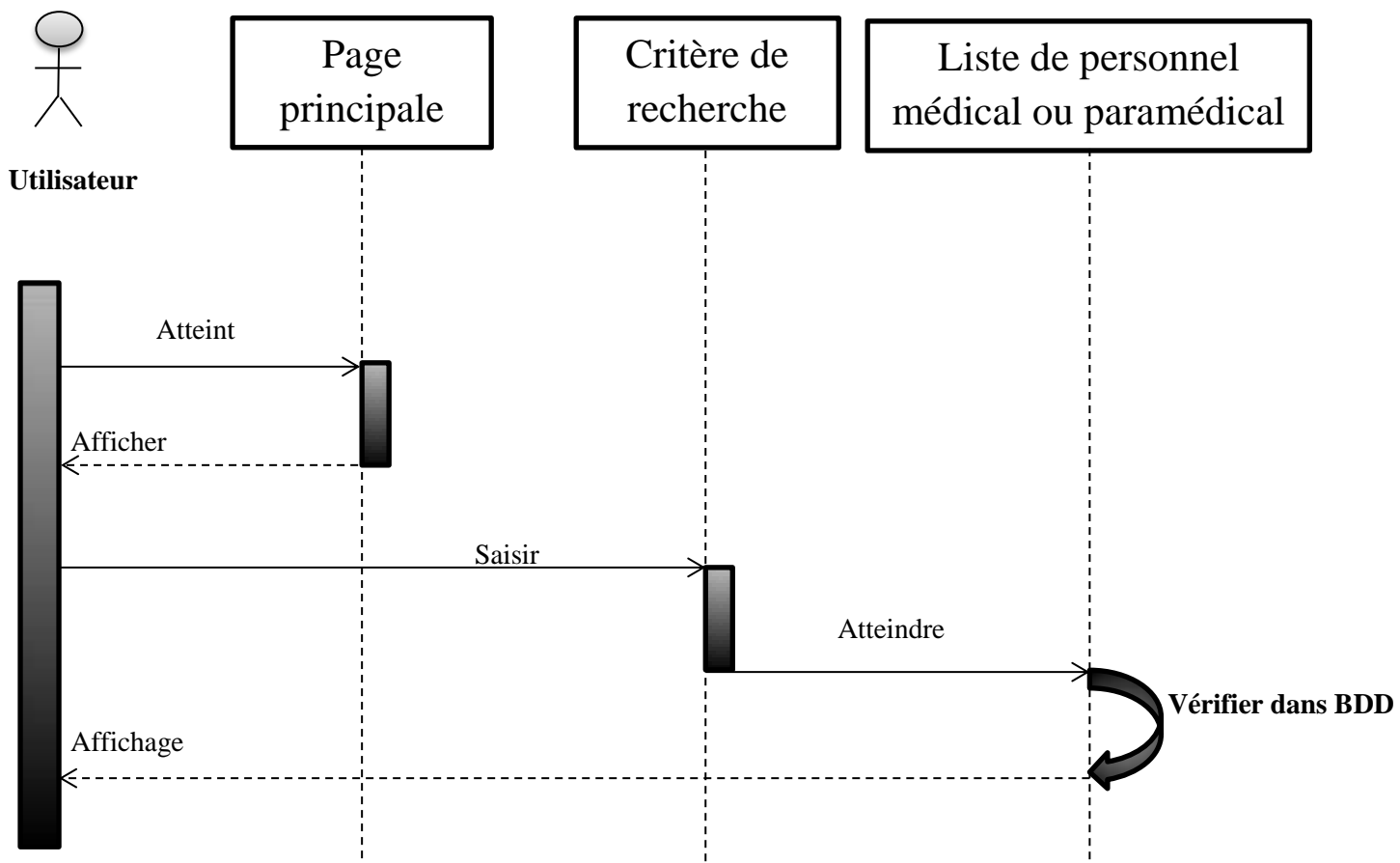


Figure II.8. Diagramme de séquence «Rechercher personnel médical ou paramédical».

IV.5.1.2. Diagramme de séquence «Appeler le personnel»

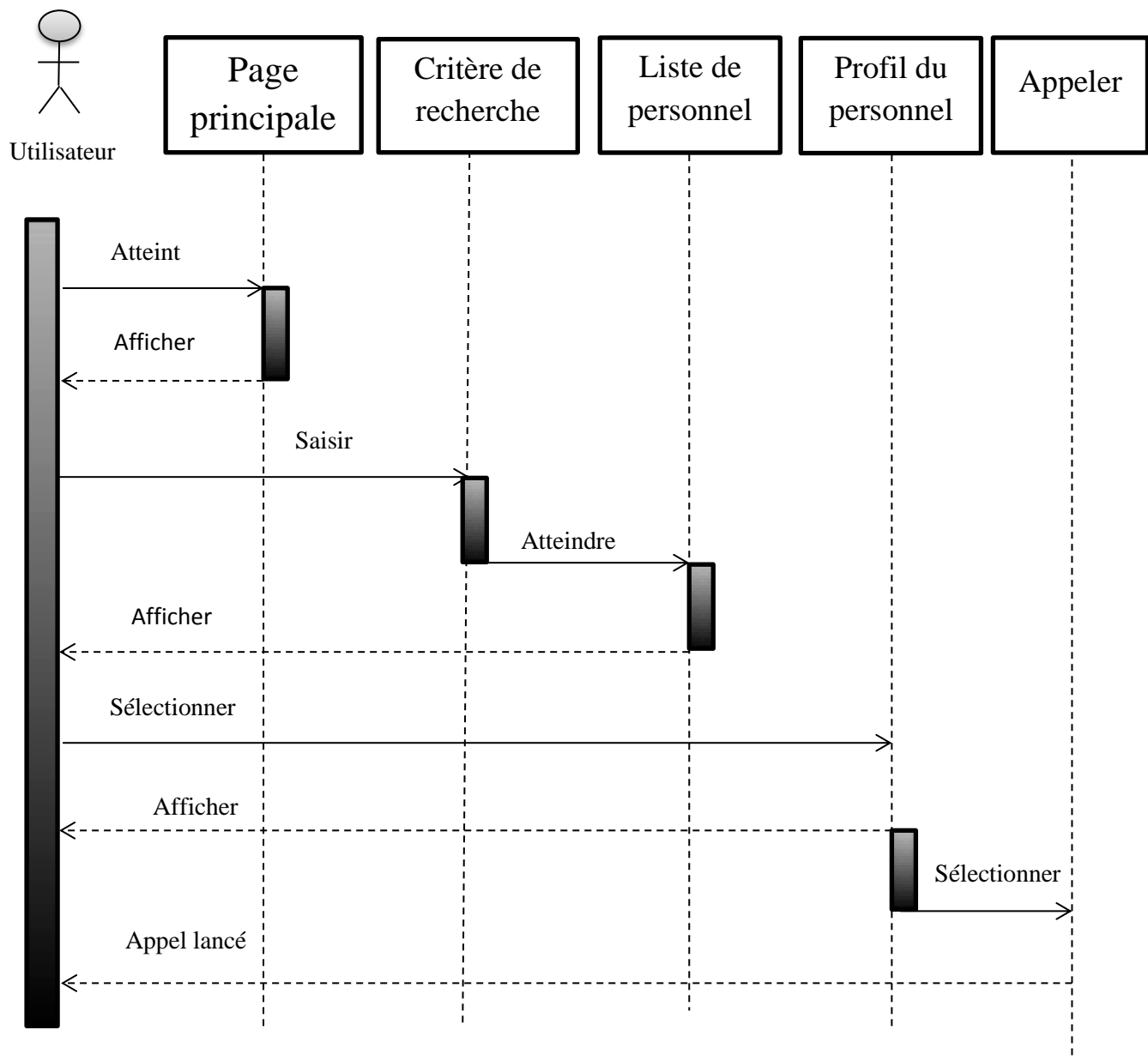


Figure II.9. Diagramme de séquence «Appeler le personnel».

IV.5.1.3. Diagramme de séquence « Prendre RDV »

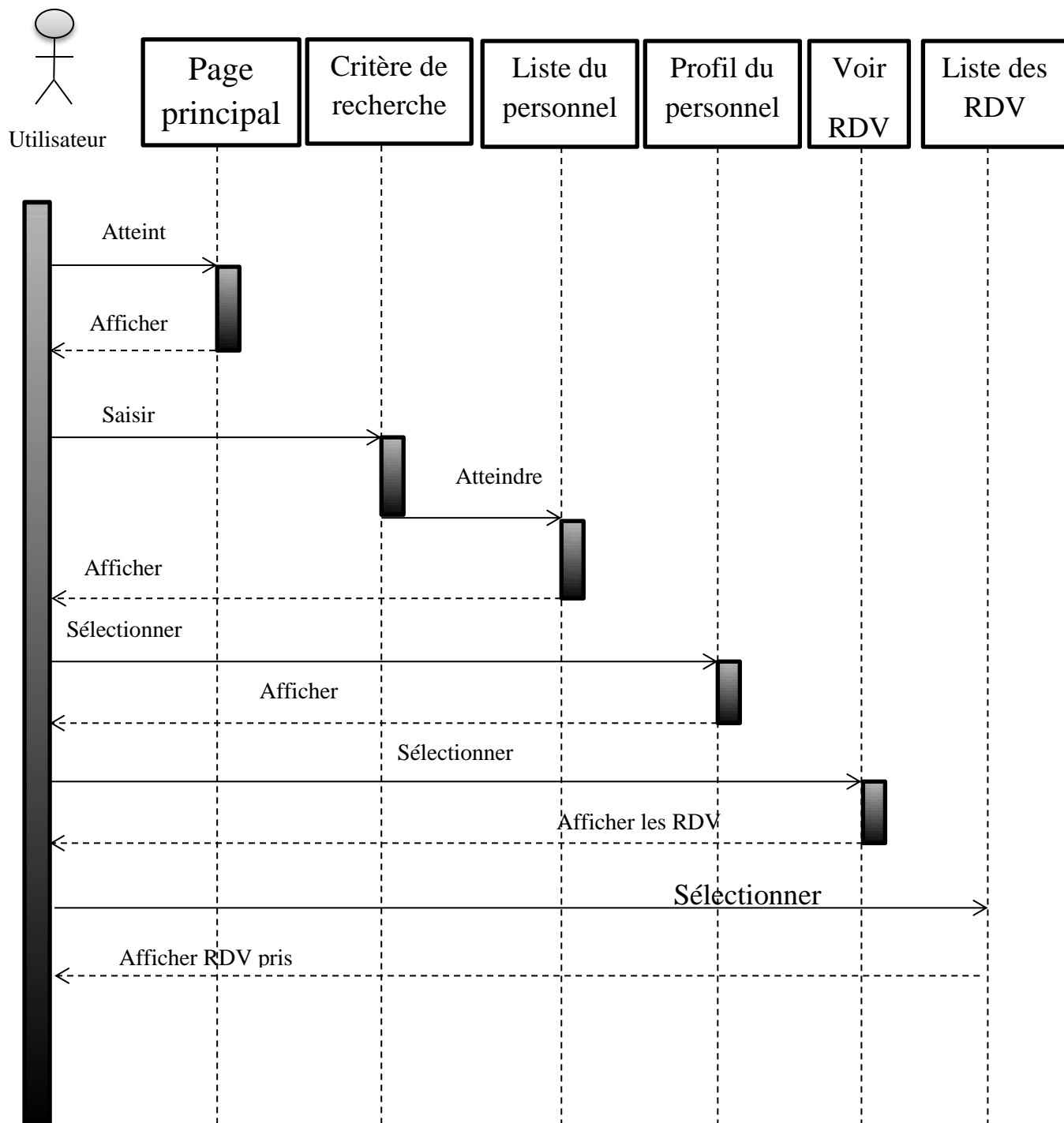
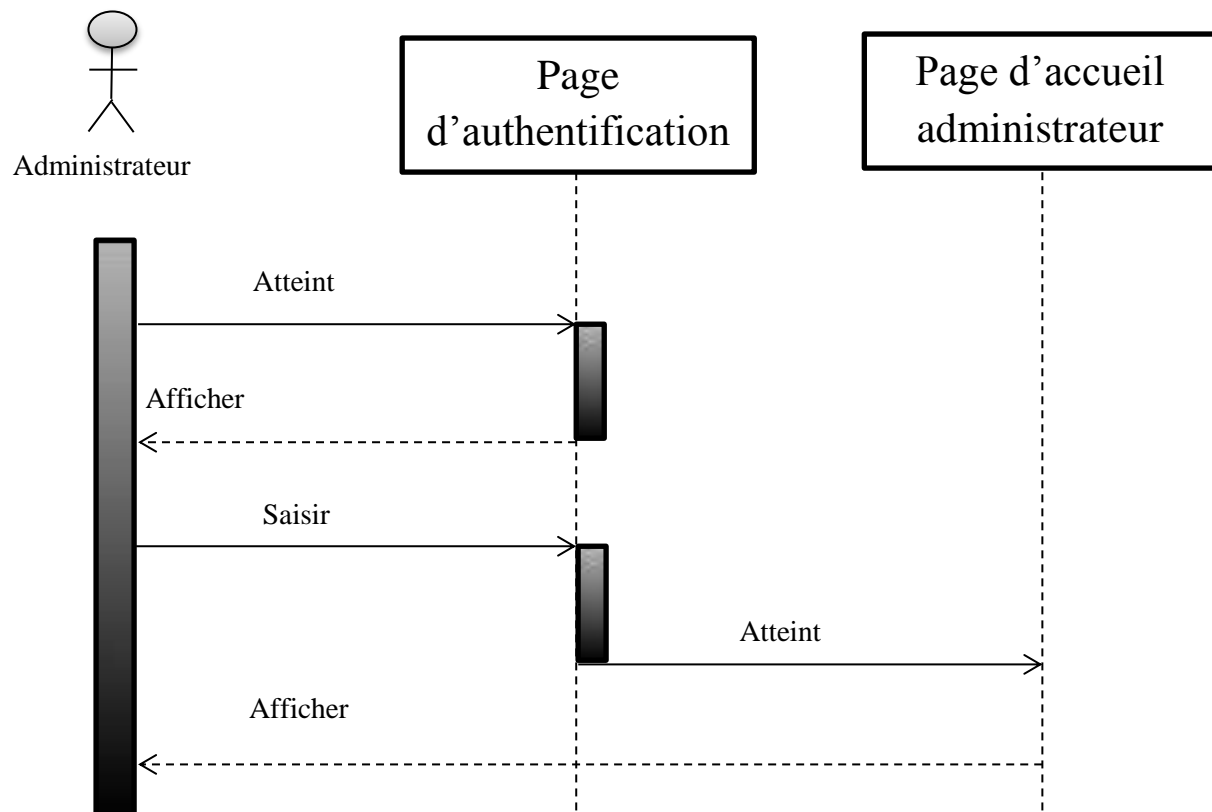


Figure II.10. Diagramme de séquence « Prendre RDV ».

IV.5.1.4. Diagramme de séquence « Authentification administrateur »

**Figure II.11.** Diagramme de séquence « Authentification administrateur ».

IV.5.1.5. Diagramme de séquence «Ajouter un personnel coté Administrateur»

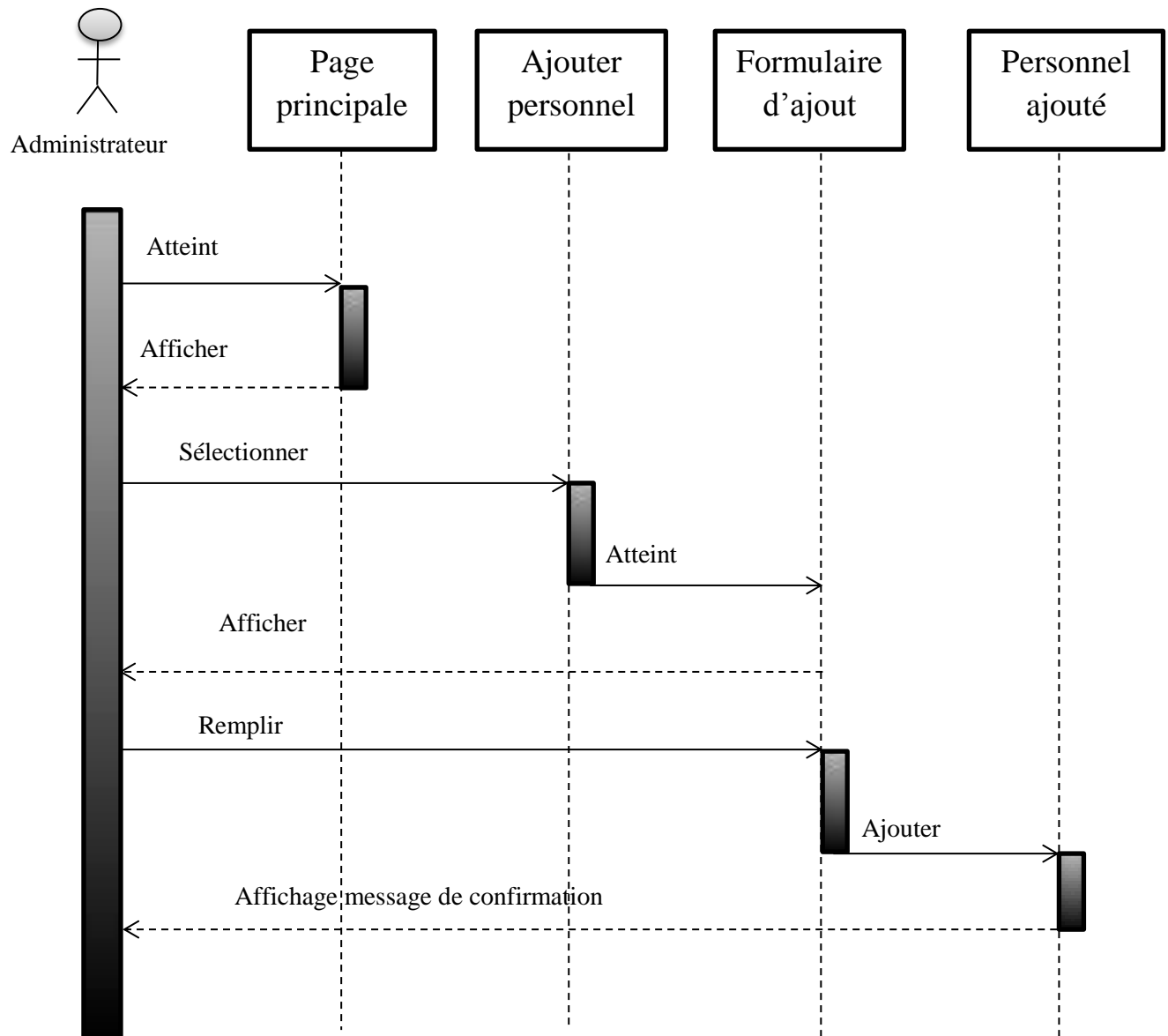


Figure II.12 : Diagramme de séquence «Ajouter personnel coté Administrateur».

IV.5.2. Le diagramme d'activité

Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles.

Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre (elles sont automatiques).

IV.5.2.1. Le diagramme d'activité «Authentification Administrateur»

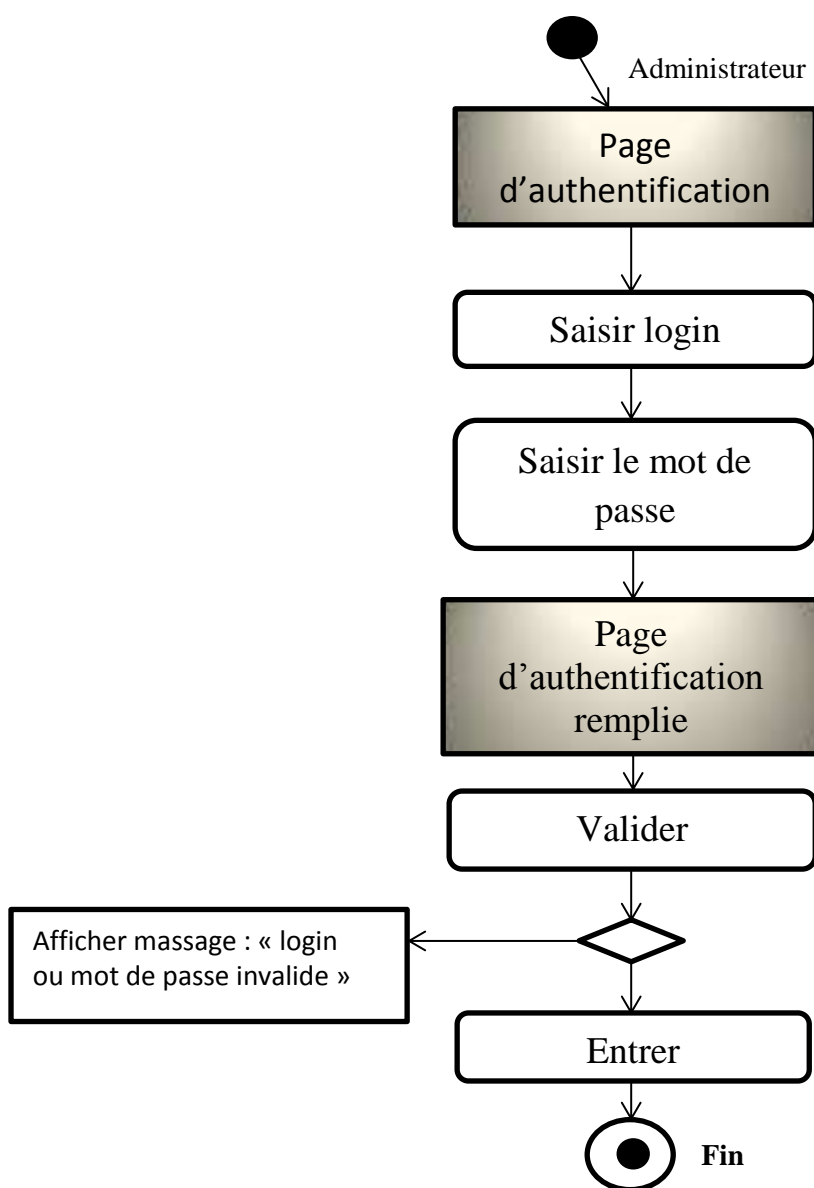
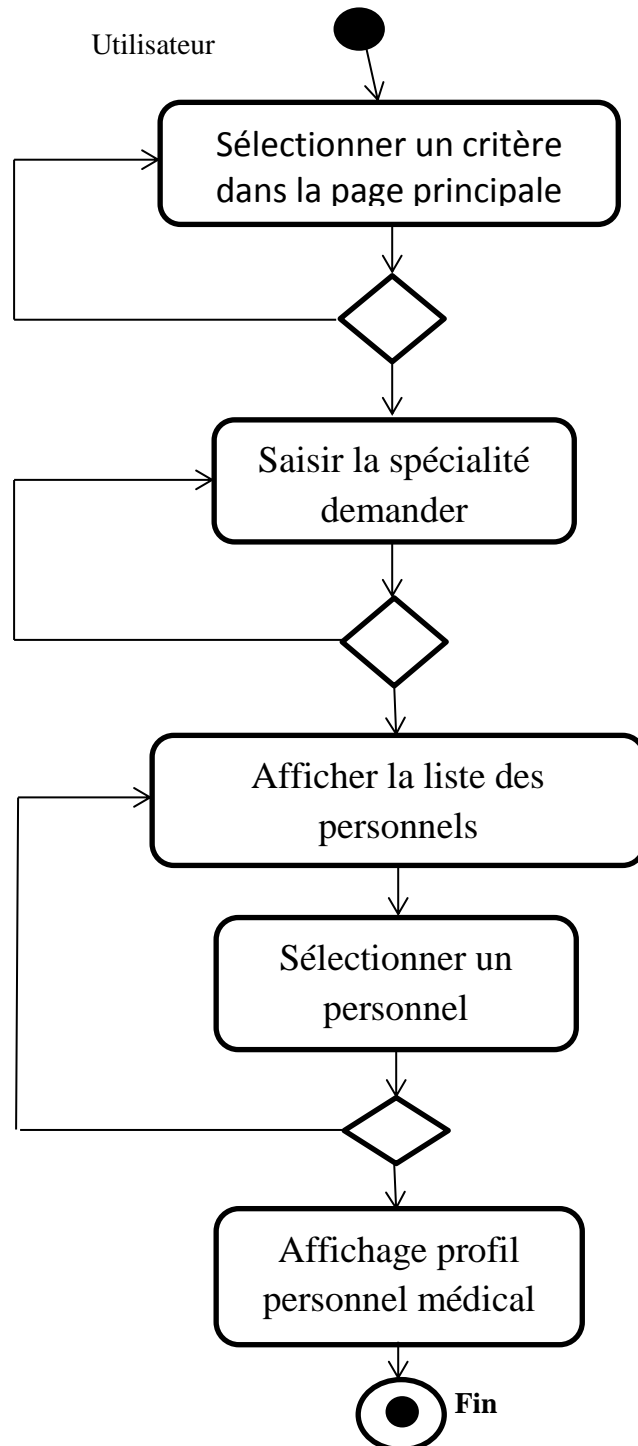


Figure II.13 : Diagramme d'activité «Authentification Administrateur».

IV.5.2.2. Diagramme d'activité « Afficher profil du personnel »

**Figure II.14:** Diagramme d'activité « Afficher profil personnel médical ».

IV.5.3. Diagramme de classe

Le diagramme de classe est l'un des diagrammes statiques d'UML. Il permet de décrire la structure d'un système informatique tout en montrant les différentes classes, leurs attributs, leurs méthodes ainsi que les relations entre eux.

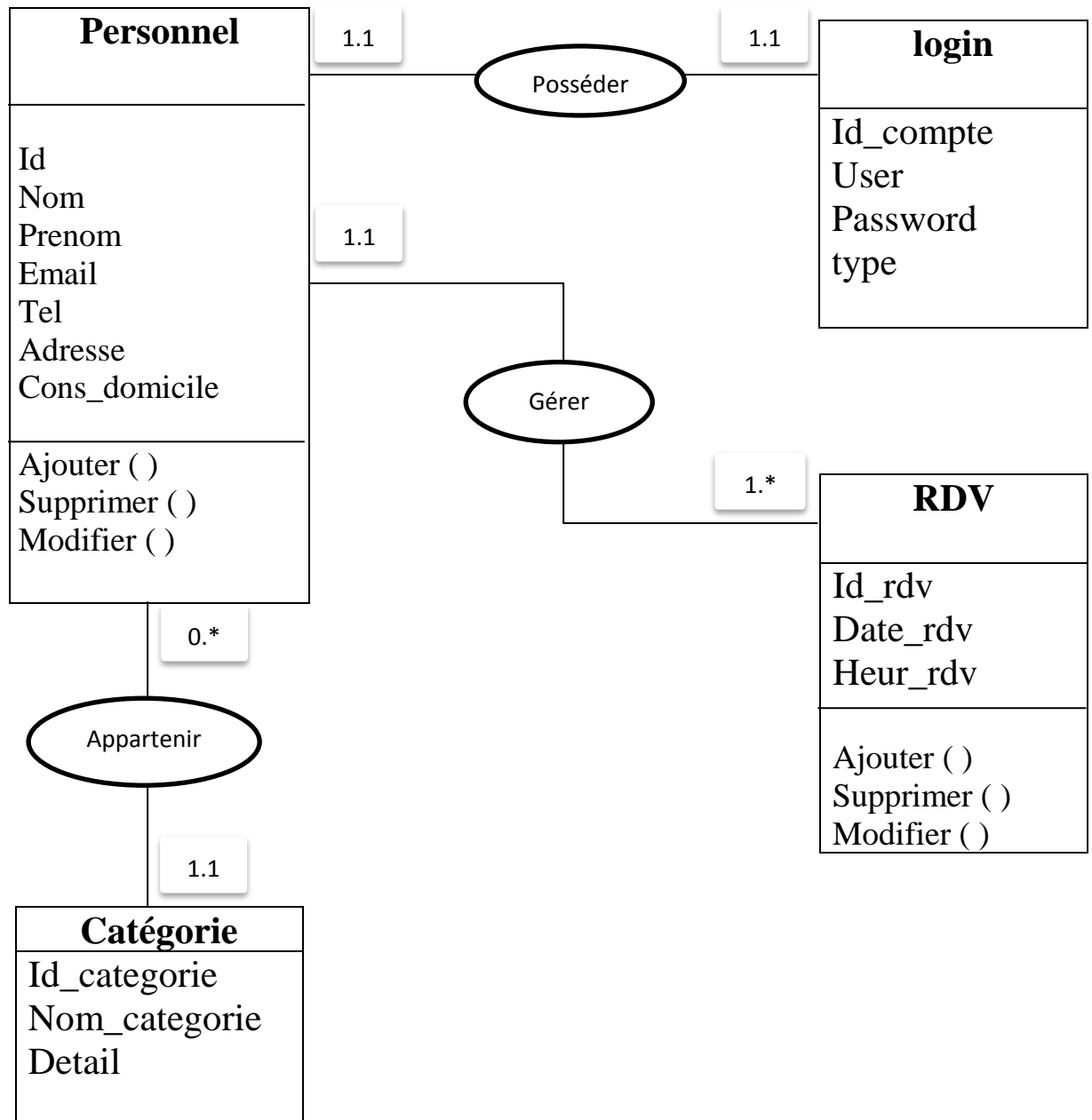


Figure II.15. Diagramme de classe globale.

V. Les tables

L'application dans tous ces cas d'utilisation manipule les données stockées dans la Base De Données, et elle contient les tables suivantes :

Champs	Types	Clé
Id-medecin	INT	Primaire
Nom	VARCHAR(30)	---
Prenom	VARCHAR(25)	---
Email	VARCHAR(50)	
Tel	VARCHAR(10)	
Adresse	VARCHAR(100)	
Cons_domicile	Boolean	
Id_compte	INT	
Id_categorie	VARCHAR(50)	Etrangère

Figure II.16: Table personnel.

Champs	Types	Clé
Id_categorie	INT	Primaire
Nom_categorie	VARCHAR(200)	
Detail	VARCHAR(200)	

Figure II.17: Table catégorie.

Champs	Types	Clé
Id_compte	INT	Primaire
User	VARCHAR(200)	
Password	VARCHAR(200)	
Type	VARCHAR(200)	

Figure II.18: Table compte.

Champs	Types	Clé
Id_rdv	INT	Primaire
Date_rdv	Date	
Heur_rdv		
Id_medecin	INT	étrangère
Etat	Boolean	

Figure II.19 : Table RDV.

VI. Conclusion

Dans ce chapitre, nous avons proposé une démarche de modélisation pour développer notre application. Cette démarche est basée sur l'UML, nous avons commencé par la spécification des cas d'utilisations dans un premier temps, suivi d'une élaboration des diagrammes de séquences et d'activités ensuite les diagrammes de classes.

I. Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement et les outils qui ont servi à la réalisation de notre application

II. Environnement de travail

II.1. Matériels utilisés

- Smartphone Condor.
- PC portable Acer.
- Un point d'accès WI-FI.

II.2. Environnement logiciel

II.2.1. Android Studio

Android Studio est un environnement de développement des applications Android.

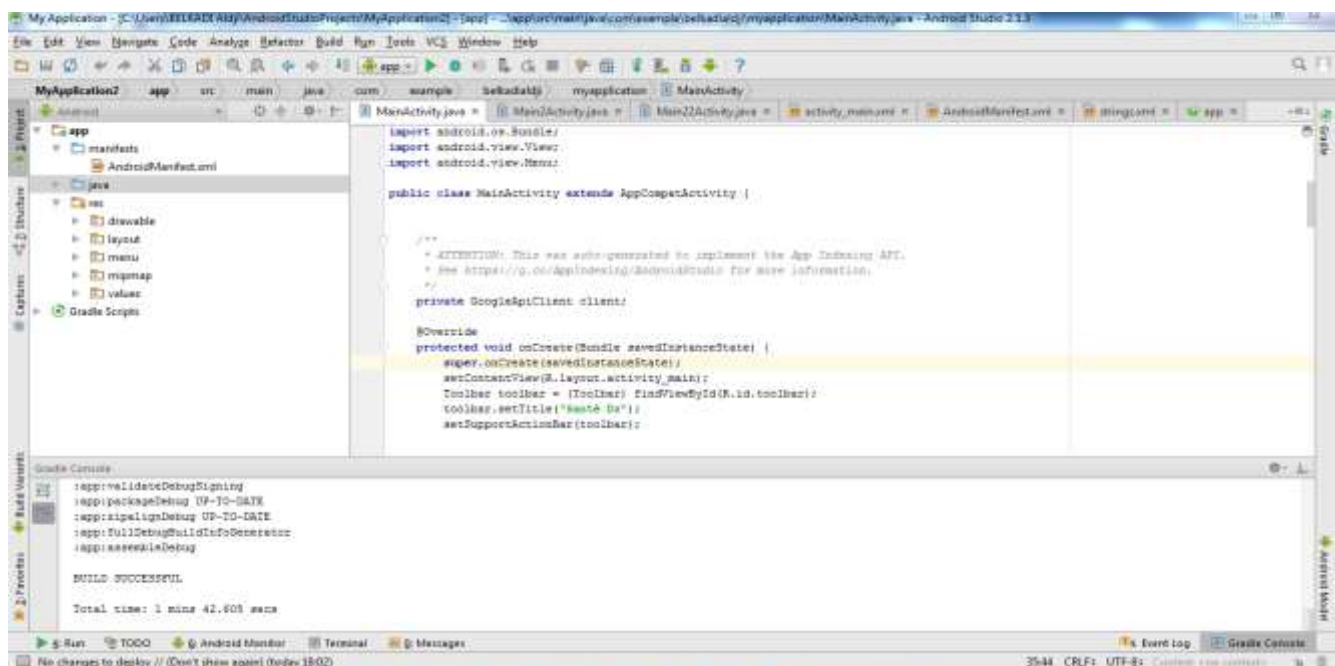


Figure III.1: Interface de l'environnement Android Studio 2.1.3

II.2.2. Software Development Kit (SDK)

Un SDK est un kit de développement, propre à chaque système d'exploitation. Il est, entre autre, utilisé pour développer des applications sous Android, il est gratuitement mis à la disposition des développeurs par Google. Le développement pour la plate-forme Android nécessite une connaissance optimale en programmation Java.

Emulateur

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel. Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth.



Figure III.2: Téléphone virtuel d'Android.

II.2.3. EasyPHP

Fut le premier package WAMP à voir le jour (1999). Il s'agit d'une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL phpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP.

II.2.4. Le serveur MySQL

MySQL est un serveur de bases de données relationnelles robuste et rapide fonctionnant sous diverses plates-formes telles que UNIX, Linux et Windows, Un serveur de bases de données stocke les données dans des tables séparées ; Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Il est basé sur le langage de requête SQL (Structured Query Language) qui est un langage standard pour le traitement des bases de données.

Le serveur MySQL offre des fonctions nombreuses et puissantes. Ses possibilités de connexions, sa rapidité et sa sécurité font de lui un serveur hautement adapté à Internet.

II.3. Les langages utilisés

II.3.1. Le langage JAVA

Le langage java est un langage de programmation orienté objet mis au point par Sun Microsystems. Sa caractéristique principale est qu'il est indépendant de toute plateforme, il est possible d'exécuter des programmes java sur tous les environnements qui possèdent une « Java Virtual Machine » (JVM). Ce concept est à la base du slogan de Sun pour java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). Sun fournit aussi gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec ce langage, ce kit est nommé JDK (Java

Development Kit). Java est caractérisée aussi par la réutilisabilité de son code ainsi que la simplicité de sa mise en œuvre.

II.3.2. XML

XML (eXtensible Markup Language, soit « Langage de balisage extensible ») est un langage de balisage définissant un format universel de représentation des données, permettant de décrire la structure hiérarchique d'un document. Le fichier à extension .xml contient à la fois les données et les indications sur le rôle que jouent ces données, ces indications (ou balises) permettent de déterminer la structure des documents.

II.3.3. Le CSS (Cascading Style Sheets)

Les feuilles de styles c'est un langage qui permet de gérer la présentation d'une page web.

Le principe des feuilles de styles consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte.

Le CSS a été mis au point afin de compenser les manques du langage HTML en ce qui concerne le design et la mise en page.

II.3.4. PHP (Hyper Text Preprocessor)

PHP est un langage de script extrêmement puissant et destiné pour le développement des applications web. PHP est l'un des langages de programmation les plus populaire. Le point fort de ce langage c'est qu'il est portable et simple à utiliser.

II.4. Choix de l'architecture de l'application

La méthode la plus répandue pour se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de mettre une sorte de service dans le milieu. MySQL est habituellement utilisé avec PHP, donc le plus évident est d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP. Nous avons codé les données dans le format JSON, afin de communiquer les données entre PHP et Android, en exploitant les options facile à utiliser construit dans les fonctions JSON dans les deux langages.

Pour mieux comprendre la procédure, nous allons la schématiser:



Figure III.3. La connexion à distance (Android /MySQL).

II.4.1. Protocole de communication

Dans notre projet, nous avons utilisé le protocole HTTP, afin de communiquer les données entre le client Mobile et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client (facilite le dispatch des fonctions).

II.4.2. Format de données communiquées

JSON (JavaScript Object Notation) est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter des informations structurées.

Un document JSON ne comprend que deux éléments structurels :

- Des ensembles de paires nom / valeur.
- Des listes ordonnées de valeurs.

Ces mêmes éléments représentent 3 types de données :

- des objets.
- des tableaux.
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou nulle.

Le principal objectif du choix d'utilisation du JSON, dans notre application, est qu'il est simple à mettre en œuvre. Comme il présente les avantages suivants :

- Facile à apprendre, car sa syntaxe est réduite et non-extensible;
- Ses types de données sont connus et simples à décrire ;
- Peu verbeux et léger, ce qui le rend bien adapté aux terminaux mobiles contrairement au langage XML qui est très verbeux.

III. Quelques interfaces

III.1. Interface Authentification d'administrateur

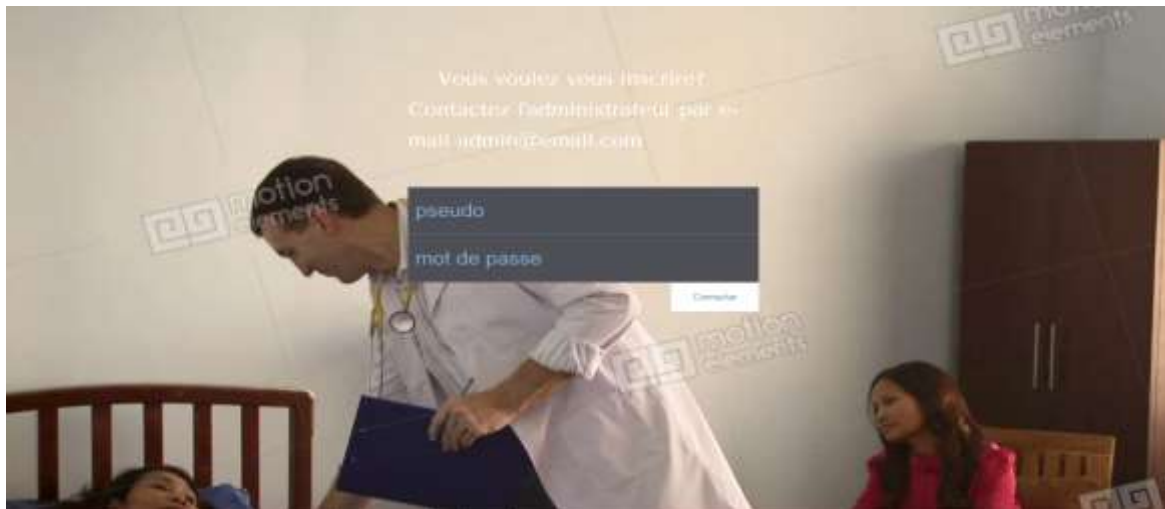


Figure III.4. Interface Authentification d'administrateur.

III.2. Interface ajout personnel



Figure III.5. Interface ajout personnel.

III.3. Interface liste personnel



Figure III.6. Interface liste des personnels.

III.4. Interface ajout R D V

Vous etes medecin?

Ajouter un R D V

Date:

Heure:

Ajouter

Figure III.7. Interface ajout R D V.

III.5. Interface d'accueil (utilisateur android)

Au premier lancement de l'application c'est l'interface principale qui sera visible, elle s'affiche lors du lancement de notre application.



Figure III.8. Interface d'accueil (Utilisateur Android)

III.6. Interface de liste des personnels (coté Android)

Quand un utilisateur choisit un critère personnel médical ou paramédical et saisi la spécialité du personnel qu'il veut alors la liste de tous les personnels s'affiche.



Figure III.9. Interface listes des personnels (coté android).

III.7. Interface envoyer un mail

Cette interface permet à l'utilisateur d'envoyer un e-mail à partir de l'application aux personnels choisis.

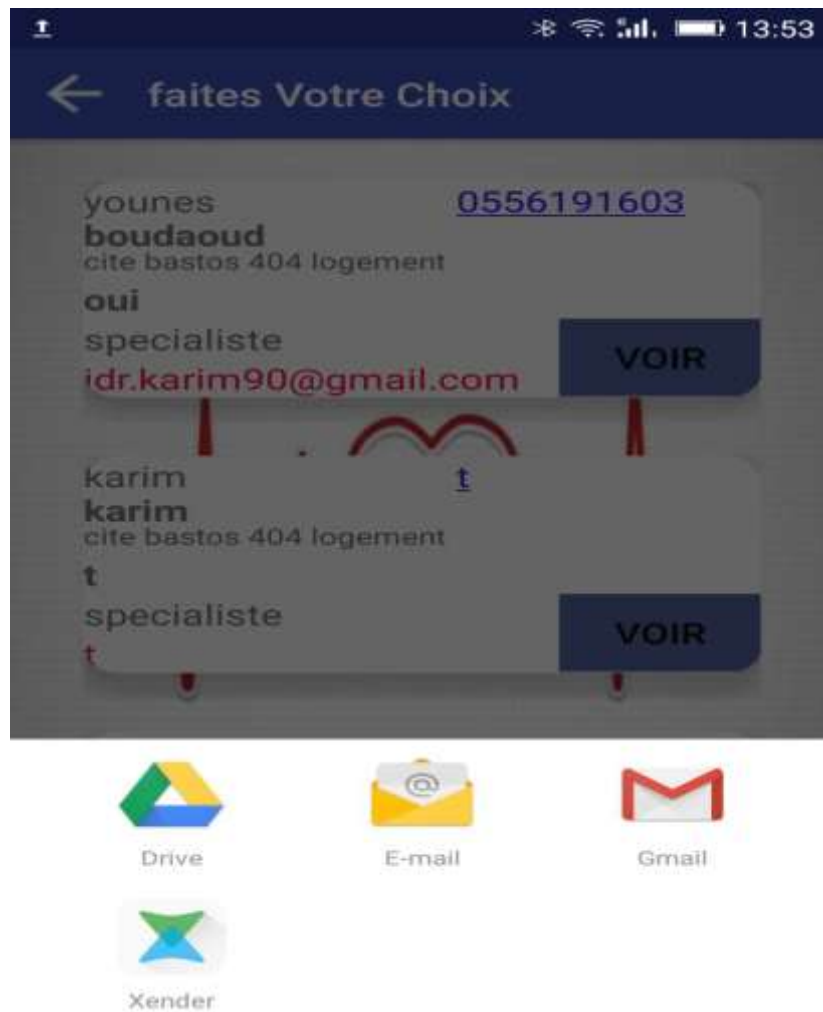


Figure III.10. Interface envoyer un e-mail.

III.8. Interface de la liste des rendez_vous



Figure III.11. Interface de la liste des RDV.

IV. Conclusion

Au niveau de ce chapitre nous avons présenté l'environnement de développement et l'implémentation de notre application. Nous avons aussi décrit quelques interfaces d'utilisation.

Conclusion générale

L'élaboration de notre travail est dans le but de concevoir une application dédiée aux terminaux mobiles disposant de la plateforme Android. Cette application permet d'accéder à l'actualité directement à partir de son Smartphone.

En partant de cela, nous avons songé à introduire une nouvelle application mobile qui s'intitule "Recherche personnel médical ou paramédical". Cette application peut pallier au manque d'information sur le personnel à consulter ainsi permettre aux patients de décrocher un rendez-vous rapidement en cas de besoin ou même d'urgence. Pour cela nous offrons à l'utilisateur une solution perspicace d'où l'idée de l'application Web, qui même si elle n'est pas visible à l'utilisateur, assure qu'il obtienne toujours l'information correcte et pertinente sans le contraindre à télécharger une nouvelle version de l'application dès qu'il y a un changement d'adresse ou de numéro de téléphone.

En effet, ce travail étant un essai, n'est donc pas un modèle unique et parfait car les perspectives d'évolution d'un tel système sont certes infinies vu le progrès technologique de plus en plus rapide et l'adaptation aussi rapide des utilisateurs à ses technologies. Parmi les perspectives envisagées pour faire évoluer ce projet sont :

- Intégration de Google Maps sous l'application android.

Bibliographie

Les systèmes d'exploitation mobiles

[1]: <https://en.wikipedia.org/wiki/IOS>

[2] : <https://fr.wikipedia.org/wiki/BlackBerry>

[3] : www.futura-sciences.com/tech/definitions/informatique-symbian-1960/

[4] : https://fr.wikipedia.org/wiki/Windows_Mobile

[5] : <http://dspace.univ-tlemcen.dz/bitstream/112/6824/1/Developpement-d'une-application-mobile.pdf>

[6], [7], [8]: Cours PROGRAMMATION ANDROID présenter par monsieur Ahcène Bounceur
Université de Bretagne Occidentale.

[9] : https://fr.wikipedia.org/wiki/Open_Handset_Alliance

[10] : <https://realitygaming.fr/threads/lavancee-technologique.466636/>

Implémentation et test

<http://developer.android.com>

<http://www.androidhive.info>

<https://openclassrooms.com/courses/creez-des-applications-pour-android>

Livres

Créer des applications Android (Auteurs : Philippe Lacomme, Raksmei Phan)

L'art du développement Android (Edition PEARSON)

Mémoires

Conception et Développement d'une Application mobile sous la Plateforme Android, mémoire de fin d'études de l'université de Tunisie.

SAIBI et TOUMERT Conception et réalisation d'une application mobile avec interconnexion à une base de données multimédia à distance dans le domaine de la médecine avancée.