



République algérienne démocratique et populaire
Université Abou Baker Belkaid-Tlemcen
Faculté de technologie

Département de Génie Electrique et Electronique

Filière : Génie Industriel

Projet de Fin d'Etudes

Master : Génie Industriel

Option : Ingénierie des systèmes

Intitulé :

«DÉVELOPPEMENT D'UNE APPLICATION DE GÉOLOCALISATION DES MÉDECINS DE LA WILAYA DE TLEMCEM SOUS ANDROID.»

Présenté par :

BEKKAYE Hadj Brahim

MEZIANE Souaad

Jury :

Encadreur : MALIKI Fouad

Président : SARI Zaki

Examineur : HASSAM Ahmed

Examineur : BENSMAINE Abderahmene

MAA

Professeur

MCB

MCB

ESSAT

UABBT

UABBT

UABBT

Année Universitaire : 2016/2017

Dédicace :

Nous dédions ce modeste mémoire à :

Nos chers parents : Que nulle dédicace ne puisse exprimer ce que nous leurs devons, pour tous leurs sacrifices, leur bienveillance, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études. Que ce travail soit témoignage de notre profond amour et notre grande reconnaissance « Que Dieu vous garde ».

Nos chères sœurs et nos chers frères : pour leurs encouragements permanents, et leur soutien moral, Nous leur dédie ce modeste travail en témoignage de notre grand amour et notre gratitude infinie.

Toutes nos Familles : à tous les familles BEKKAYE & DADDA et MEZIANE & SACI, pour leur soutien tout au long de mon parcours universitaire, Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible.

Tous nos amis : Pour leur aide et leur soutien moral durant l'élaboration du travail de fin d'études.

Remerciement :

Avant tout, nous remercions le bon DIEU de nous avoir aidés à accomplir ce modeste travail.

Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances à :

Notre encadreur : Mr MALIKI Fouad pour son disponibilité, ses conseils, son orientation et ses encouragements tout au long de notre recherche.

Mr DADDA Mustapha qui a suivi chaque étape de ce travail, et nous a aidé surtout dans la réalisation de l'application (Designer).

Membres du jury Mr SARI Zaki, Mr HASSAM Ahmed et Mr BENSMAINE Abderrahmane qui ont accepté d'évaluer et d'examiner ce travail.

Nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous ne saurions terminer ces remerciements sans n'y associer toute personne qui a participé de près ou de loin à l'exécution de ce modeste travail.

Table des matières

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

CHAPITRE 1 : Généralités sur les applications mobiles

Introduction :.....	3
I. Les applications mobiles :.....	3
1. Définition :.....	3
2. Les types d'applications :.....	4
3. Caractéristiques des applications mobiles :.....	5
II. Les différentes plateformes mobiles :.....	6
1. Android :.....	6
2. IOS :.....	7
3. Windows Phone 7 :.....	7
4. BlackBerry OS :.....	8
5. Les autres systèmes d'exploitation :.....	8
III. Le système d'exploitation Android :.....	10
1. La naissance d'Android :.....	10
2. L'historique des versions d'Android, une véritable évolution :.....	10
3. Architecture logicielle :.....	14
4. Kit de développement :.....	15
IV. La Géo-localisation :.....	16
1. Google Maps :.....	17
V. Les bases de données :.....	17
1. Qu'est-ce qu'une base de données ?.....	17
2. Qu'est-ce qu'un SGBD ? (Data Base Management System).....	19
Conclusion :.....	24

CHAPITRE 2 : Outils utilisés et conception UML

Introduction :.....	26
I. Problématique :.....	26
II. Environnement de travail :.....	27
1. SDK :.....	27
2. Android Studio :.....	27
3. Serveurs :.....	29

4. WampServer :.....	29
5. Intégration de Google Map :.....	30
6. Photoshop :.....	31
III. Besoins et conception :.....	31
1. Les besoins non fonctionnels :.....	31
2. Les besoins fonctionnels :.....	32
IV. Conception UML de notre application :.....	32
1. Définition de l'UML :.....	32
2. Diagramme de cas d'utilisations :.....	33
3. Diagrammes de classes :.....	34
4. Diagrammes d'activité :.....	35
Conclusion :.....	36

CHAPITRE 3 : Modélisation et réalisation

Introduction :.....	38
I. Objectif :.....	38
II. Description des interfaces de l'application :.....	38
1. Logo de l'application :.....	38
2. Capture d'écrans des activités :.....	38
3. Navigabilité de l'application entre les activités :.....	43
Conclusion :.....	49
Conclusion générale :.....	50
Bibliographie :.....	51

Liste des figures

Figure 1.1 : Le processus de reconnaissance par empreinte digitale.....	3
Figure 1.2 : Logo du système Android.....	6
Figure 1.3 : Logo du système iOS.....	7
Figure 1.4 : Logo du système Windows Phone 7.....	8
Figure 1.5 : Logo du système BlackBerry OS.....	8
Figure 1.6 : Situation d'Android début février 2017.....	14
Figure 1.7 : Architecture logicielle de l'Android.....	14
Figure 1.8 : Première vue d'un SGBD.....	20
Figure 1.9 : Composants d'un environnement base de données.....	21
Figure 1.10 : Serveur et clients de MySQL.....	24
Figure 2.1 : SDK Manager.....	27
Figure 2.2 : Logo de la plateforme Android Studio.....	28
Figure 2.3 : Structure d'un projet Android.....	29
Figure 2.4 : Logo WampServer et ses comprenant.....	29
Figure 2.5 : Google Play Services.....	30
Figure 2.6 : La clé API de Google Map.....	30
Figure 2.7 : La clé API au fichier AndroidManifest.....	30
Figure 2.9 : Le code pour l'activity_main.xml.....	31
Figure 2.10 : Diagrammes des cas d'utilisations.....	33
Figure 2.11 : Diagrammes de classe.....	34
Figure 2.12 : Diagramme d'activité -Ajouter cabinet.....	35
Figure 2.13 : Diagramme d'activité - Recherche médecin -.....	36
Figure 3.1 : Logo.....	38
Figure 3.2 : Activité démarrage.....	39
Figure 3.3 : Activité de connexion.....	39
Figure 3.4 : Activité de registration.....	39
Figure 3.5 : registration plus détails sur le cabinet.....	39
Figure 3.6 : Activité principale.....	40
Figure 3.7 : Liste des médecins d'une spécialité.....	40
Figure 3.8 : fiche d'information.....	40
Figure 3.9 : Navigation drawer.....	40

Figure 3.10 : Numéros utile.....	41
Figure 3.11 : Activité recherche.....	41
Figure 3.12 : A propos de.....	41
Figure 3.13 : Activité Map.....	41
Figure 3.14 : Modifier information.....	42
Figure 3.15 : Activité démarrage.....	43
Figure 3.16 : Activité de connexion.....	43
Figure 3.17 : Activité de registration.....	43
Figure 3.18 : Activité principale.....	43
Figure 3.19 : Activité de connexion.....	43
Figure 3.20 : registration plus détails sur le cabinet.....	43
Figure 3.21 : registration plus détails sur le cabinet.....	44
Figure 3.22 : Choix d'horaires.....	44
Figure 3.23 : Choix de jour.....	44
Figure 3.24 : Choix d'attestation.....	44
Figure 3.25 : Activité principale.....	45
Figure 3.26 : Liste des médecins d'une spécialité.....	45
Figure 3.27 : Appeler le médecin.....	45
Figure 3.28 : Activité Map.....	45
Figure 3.29 : fiche d'information.....	45
Figure 3.30 : fiche d'information.....	46
Figure 3.31 : Appeler le médecin.....	46
Figure 3.32 : Activité Map.....	46
Figure 3.33 : Navigation drawer.....	47
Figure 3.34 : Activité principale.....	47
Figure 3.35 : Numéros utile.....	47
Figure 3.36 : Activité recherche.....	47
Figure 3.37 : A propos de.....	47
Figure 3.38 : Navigation drawer.....	48
Figure 3.39 : Activité de registration.....	48
Figure 3.40 : Navigation drawer.....	48
Figure 3.41 : L'affiche de message.....	48
Figure 3.42 : Modifier information.....	48

Liste des tableaux

Tableau 1.1 : Une comparaison entre les systèmes d'exploitation mobile.....9

Tableau 1.2 : Les versions de l'Android.....13

Liste d'abréviations

BD : Base de Données.

SGBD : Système de Gestion de Base de Données

SDK: Software Development Kit.

OS: Operating System.

IOS: Internetworking Operating System.

API: Application Programming Interface.

HTML: Hypertext Markup Language.

CSS: Cascading Style Sheets.

AIDL : Android Interface Definition Language.

UMTS : Universel Mobile Télécommunications System.

Introduction générale

Avec l'indispensabilité de la communication et de l'information dans notre vie quotidienne et grâce à la généralisation des téléphones portables tactiles à écrans large ainsi qu'au développement des logiciels et des réseaux, la recherche d'information est accessible du bout des doigts par l'intermédiaire de téléphone ou pads à tout moment et en tous lieux.

Il est important pour tout individu de se tenir au courant de ce qui se passe autour de lui et d'avoir accès aux informations dont il a besoin ,quand il en a besoin plus facilement et plus rapidement.

Les applications mobiles sont capables de satisfaire un large éventail de besoins. Elle avoir une rapidité d'accès à l'information ainsi que la possibilité de profiter de plusieurs fonctionnalités novatrices, ludique et intuitives qui repoussent les limites de l'expérience utilisateur comme : l'appareil photo, vidéo, le GPS... en plus elles offrent une interface plus spécifique et plus harmonieuse qui les rendre plus aisé que les sites mobiles dans leur utilisation. Pour cette raison l'idée de développement une application devient utile et conviviale.

Dans le domaine de la santé, le patient a des difficultés pour chercher un médecin ou un spécialiste pour faire une visite générale ou suivre la santé de son Corps, ainsi que le médecin besoin de former les gens de sa position ou les horaires de travail... .Donc il est important pour des corps médicaux de se tenir proche de ses clients.

Il s'agit au cours de notre projet d'étudier et de développer une application sous Android comme une solution simple et innovante permettant de contacter et localiser un médecin dans la spécialité demandée en quelques clics.

Ce mémoire est organisé en trois chapitres, le premier présente les différentes notions utilisées dans le développement des applications mobiles à savoir, le système d'exploitation Android, la géo-localisation et les bases de données.

Dans le deuxième chapitre nous présentons nos objectifs ainsi que la conception UML réalisée.

Ce mémoire est clôturé par un troisième chapitre dans lequel nous présentons l'application de géo-localisation des médecins de la wilaya de Tlemcen réalisée.

CHAPITRE 1 :
Généralités sur les applications mobiles

Introduction :

Avec l'évolution de la stratégie mobile et la tendance actuelle à l'utiliser de plus en plus dans la vie quotidienne, nombreux sont les secteurs d'activité à l'exploiter pour se développer.

La géo-localisation est un procédé permettant de positionner un objet (une personne, un véhicule...) sur un plan ou une carte à l'aide de ses coordonnées géographiques.

Une base de données est une collection de données organisées de façon à être facilement accessibles, administrées et mises à jour.

Dans ce chapitre nous présentons les différentes notions de l'environnement Android et ses fonctionnalités ainsi que la géo-localisation et les bases de données.

I. Les applications mobiles :

1. Définition :

Une application mobile est une application embarquée qui est sensé s'exécuter sur une plateforme mobile à l'instar des Smartphones et les tablettes.

L'application doit être téléchargée par l'utilisateur pour devenir exécutable à partir du système d'exploitation du téléphone exactement comme un logiciel ordinaire qui s'installe sur un ordinateur classique. Les applications mobiles ne sont pas transférables d'un système d'exploitation à l'autre en raison du langage informatique utilisé : si vous téléchargez une application Android, vous ne pourrez l'utiliser que sur un téléphone portable Android.

Reflète du succès commercial et technique de l'iPhone, une grande majorité des applications mobiles furent créées à l'origine pour le téléphone mobile d'Apple. Cependant, les applications Android se sont fortement développées depuis les années 2010 / 2011 et dépassent désormais en nombre d'installation les applications iPhone.

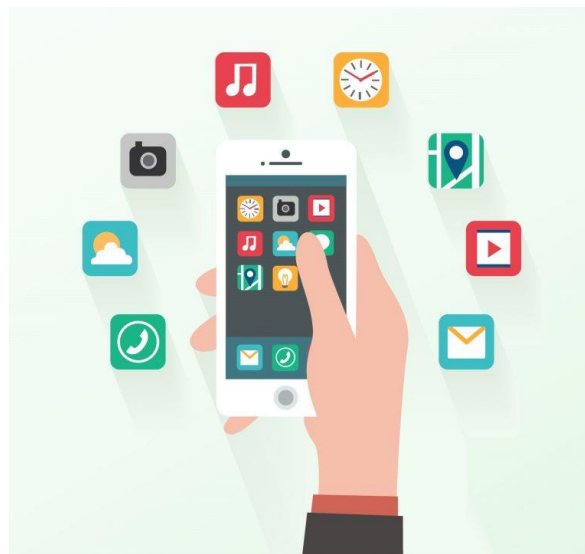


Figure 1.1 : Le processus de reconnaissance par empreinte digitale.

Chez les mobinautes, l'application mobile est similaire à un site internet pointu en raison de sa connexion à internet, de plus l'interface du site et de l'application mobile s'avèrent identiques sauf que l'application demeure fondamentalement défini comme un logiciel. En ce sens, les applications mobiles se regroupent en plusieurs séries suivant des critères basiques :

a) Applications fonctionnant sans internet : Appelées applications indépendantes, ce sont des applications qui fonctionnent sans avoir besoin de connexion internet ou téléphonique dont on cite la liste de contacts, la calculatrice et autres.

b) Applications exigeant une connexion : Contrairement aux applications indépendantes, ces applications doivent avoir accès à l'internet pour fonctionner.

c) Applications connectées : C'est une application qui nécessite une connexion internet pour un bon fonctionnement.

d) Applications interagissant avec les autres équipements de Smartphone : Tout Smartphone dispose d'une suite d'équipements pointus davantage que les téléphones portables ordinaires et ces équipements sont en interaction permanente avec certaines applications. On en cite le scan de code barre ou code pour savoir le prix et les caractéristiques d'un produit vendu sur les hypermarchés.

e) Applications interagissant avec d'autres mobinautes : La génération Y est familière à se connecter sur internet et le téléphone mobile ce qui justifie l'importance des applications qui renforcent les liaisons entre les mobinautes au lieu de se connecter passivement sur les réseaux sociaux.

2. Les types d'applications :

Techniquement parlant, il y a trois types d'application mobile que tout utilisateur peut rencontrer :

a) Application native : Il s'agit d'application conçue pour une grande partie de systèmes d'exploitation fiables par les Smartphones en se référant à un langage particulier à chacun d'eux.

Ce mode d'application est accessible seulement sur les plateformes d'applications suivant ses particularités et ses formules. Ces plateformes retirent 25% du prix de vente pour une application native payante.

Le développement de l'application native nécessite le recours à la mémoire du Smartphone sans omettre les options reliées au système d'exploitation en question. De cette façon, le résultat se résume dans l'aboutissement à des applications mobiles avec des fonctions plus professionnelles, développées et performantes au même niveau que les applications en HTML5/CSS3 et les applications hybrides.

Le souci des applications natives est que les utilisateurs doivent avoir un système d'exploitation mobile donné pour qu'ils puissent les utiliser. Pour assurer un usage plus exponentiel de ces applications mobiles, il faut penser à lancer la même application mobile compatible à tout système d'exploitation mobile.

b) Application web : Toute application conçue avec HTML et CSS de plus opérationnelle sur navigateur internet pour un Smartphone est appelée application web.

Peu importe la marque de votre Smartphone, vous pouvez accéder à l'application web par le biais de son navigateur et donc vous n'avez pas besoin de la télécharger. Vu qu'elle ne tient pas en compte les divergences persistantes entre les systèmes d'exploitation et les marques de Smartphone, l'application web manque d'ergonomie et de plus elle ne se sert pas de la mémoire du Smartphone ce qui la place en infériorité par rapport à l'application native.

c) Application hybride : Il s'agit d'une application mobile qui fusionne entre les caractéristiques de l'application web (développement en HTML 5) et celles de l'application native. De cette manière, l'application mobile sera accessible sur toutes les plateformes d'application. Ce type d'application mobile minimise les charges et la durée de son développement même si cela sera au détriment du perfectionnement et de la qualité qui caractérise l'application native. Notons que les applications hybrides sont accessibles exclusivement sur iPhone et Android.

3. Caractéristiques des applications mobiles :

a) Une application mobile est plus rapide : En effet, sauf si elle est très mal programmée (ce qui peut arriver), une application native va exploiter au mieux les capacités du téléphone. En plus, en utilisant les composants graphiques du téléphone (boutons, barre de titre, onglets...), l'interface est plus facile à appréhender pour l'utilisateur qui retrouve rapidement ses marques.

b) Une application offre plus de possibilités : Accéder aux photos, à l'agenda, aux contacts, proposer de la réalité augmentée, réagir aux mouvements du téléphone, lire un code barre : autant de fonctionnalités qui ne sont possibles qu'avec une application mobile.

c) C'est plus difficile à développer : On ne va pas dire le contraire. C'est pourquoi on vous conseille de faire appel à des professionnels. Ce qui ne veut pas dire que le budget sera démesuré...

d) Il faut développer pour chaque plateforme : Là encore, c'est tout à fait vrai. Pour s'adapter à l'ergonomie de chaque plateforme, exploiter au mieux les fonctionnalités, il faut développer pour chaque type de téléphone (iPhone, Android...). Cependant, décliner une application vers une nouvelle plateforme coûte moins cher, car une bonne partie de l'application peut être réutilisée.

e) On gagne plus avec une application mobile: Si vous avez comme projet de vendre votre application, c'est effectivement plus aisé qu'avec une Web App, vente de l'application sur l'AppStore ou l'Android Market, vente d'articles ou de fonctionnalités à l'intérieur de l'application, tout ceci est possible très simplement.

f) Le déploiement est plus compliqué, plus long : Apple impose une validation des applications avant publication. C'est vrai. Il n'y a cependant rien de tel sur AndroidMarket⁶, et les mises à jour sont dans tous les cas signalées aux utilisateurs, ce qui remet un coup de projecteur dessus. Enfin, pour les entreprises, il est possible de déployer ses applications en interne, même sur iPhone/iPad, sans passer par l'AppStore.

g) Pas besoin d'être connecté pour utiliser mon application : Cela dépend de votre application, mais effectivement, il n'est pas nécessaire de se connecter à l'internet pour lancer

l'application. Par contre, si celle-ci doit accéder à des informations depuis un serveur (coordonnées des boutiques, dernières actualités...), il est évident que la connexion est toujours nécessaire.

II. Les différentes plateformes mobiles :

Aujourd'hui le marché mondial des téléphones mobile est dominé par quatre grandes entreprises de technologie Smartphone qui sont Google, Apple, RIM et Microsoft qui développent respectivement les systèmes d'exploitation Android, iOS, BlackBerry OS et Windows Phone 7.

Pour connaître le leader dans le marché des Smartphones et déterminer le système le plus répondant aux besoins de notre application, nous allons présenter brièvement chacun de ces systèmes et décrire leurs avantages et inconvénients :

1. Android :

Android est un système d'exploitation ouvert (Open Source) pour tablettes tactiles, terminaux mobiles et TV connectées. Il a été conçu en 2007, par la société Android, une start-up rachetée par Google. C'est un système d'exploitation fondé sur un noyau Linux Disponible grâce à une licence Apache.

Android inclut tous les utilitaires requis par un constructeur pour le mettre en œuvre dans un téléphone portable. Il est proposé à tous les fabricants de téléphones mobiles pour faciliter son adoption. Il a été conçu pour intégrer les applications Google comme Gmail, Google Map, Google Agenda, Google Talk ou encore You Tube.

Android est un système presque équivalent à IOS, tant en fonctionnalités qu'en applications. Là où Apple cherche la cohérence et la sécurité, Android offre plus de choix et de personnalisation.

Pour développer une application mobile sur ce support, le langage de programmation utilisé sera Java. Le « kit de développement » ou SDK (Software Développement Kit) autrement dit les outils de développement utilisés seront le SDK Android. On peut également utiliser le langage C++ avec le NDK (Native Développement Kit).



Figure 1.2 : Logo du système Android.

2. IOS :

Anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone. Reconnu pour sa fluidité, son ergonomie et son intuitivité : c'est le système d'exploitation le plus abouti à ce jour. Il dispose du portail App Store, qui avec un catalogue de plus de 200 000 applications, s'est imposé comme une référence parmi les kiosques d'applications mobiles.

Après Android, iOS est le deuxième système d'exploitation le plus répandu. Il équipe uniquement les « iPhone », c'est-à-dire les Smartphones de la marque Apple. Le système d'exploitation mobile d'Apple a toujours été cité en exemple pour sa grande simplicité d'utilisation. Il se compose en effet principalement de multiples écrans d'accueil qui abritent les applications. Il est possible de créer des dossiers d'applis, mais pas de placer des widgets ou un quelconque autre moyen de visualiser des informations directement sur les écrans d'accueil. Après avoir fait de la résistance, Apple a fini par proposer un centre de notifications — bien après Android —, plutôt bien fichu, ainsi qu'une barre de raccourcis pour accéder rapidement aux connectivités, à la lampe torche, à l'APN, etc. On apprécie tout particulièrement la fonction de recherche universelle. Comme son nom le laisse entendre, elle va fureter partout dans le système d'exploitation et se révèle vite indispensable pour les utilisateurs les plus exigeants. Enfin, le système de multitâche s'avère lui aussi très bon, même si certains peuvent lui préférer d'autres systèmes qui montrent davantage d'applis récemment utilisées sur la même fenêtre.



Figure 1.3 : Logo du système iOS.

3. Windows Phone 7 :

Dernier né des systèmes d'exploitation, Windows Phone 7 se différencie de ses concurrents par son système de hub qui regroupe tous les éléments d'une même catégorie (contacts, photos, jeux...). Par exemple, lorsque vous vous rendez sur le blog contacts, vous pourrez avoir accès aux derniers appels, messages et photos Facebook....

Windows Phone 7 est très exigeant sur les spécifications des Smartphones : la mémoire et le processeur doivent ainsi atteindre des seuils minimum de performance. Des caractéristiques qui rendent la navigation fluide et agréable. L'interface propose également une plateforme de jeu performante avec l'accès à son service de jeu en ligne Xbox Live.

Autre apport non négligeable, Windows Phone 7 donne accès au Pack Office (Word, Excel, Powerpoint...) : idéal pour la productivité. Seul point faible de Windows Phone 7, le

faible nombre d'applications. En effet, étant plus jeune que ses concurrents il ne dispose pas d'un catalogue aussi conséquent que celui de l'Android Market ou de l'AppStore, mais de nombreux développeurs contribuent à rattraper son retard.



Figure 1.4 : Logo du système Windows Phone 7.

4. BlackBerry OS :

La plate-forme de communication sans fil BlackBerry est développée par Research In Motion (RIM). Les solutions proposées par RIM intègrent à la fois les matériels, les logiciels et les services qui permettent un accès simple et transparent aux informations importantes, telles que messages électronique, téléphoniques, SMS ainsi qu'aux applications Internet et Intranet.

Les produits, services et technologies de RIM ont reçu de nombreuses récompenses et sont utilisées par des milliers d'entreprises dans le monde.

Longtemps réservé aux professionnels, BlackBerry a dû s'adapter à sa nouvelle clientèle, plus jeune et tombée sur le charme des claviers AZERTY, très pratiques pour l'envoi de messages écrits. D'une interface austère et dédiée à l'email (que l'on peut envoyer et recevoir en instantané), BlackBerry OS est devenu au fil du temps agréable à utiliser et de plus en plus complet. Le système regorge de bonnes idées à commencer par la possibilité de séparer son espace de travail de sa vie privée sur un seul et même téléphone. Le magasin d'application maison, l'App World, continue à se remplir, mais la force de l'OS est sa compatibilité avec les programmes Android, qui ne nécessitent qu'une petite optimisation de la part des développeurs.



Figure 1.5 : Logo du système BlackBerry OS.

5. Les autres systèmes d'exploitation :

D'autres Smartphones ne proposant pas les systèmes d'exploitations évoqués ci-dessus existent, mais représentent encore dans leur totalité une partie extrêmement infime des ventes de Smartphones au niveau mondial, de l'ordre de 14 millions en 2014, sur un total de près de

1,25 milliards de Smartphones. Parmi les près de 20 systèmes d'exploitation mobiles existants, on peut citer Linux (OS open source) et le plus récent Firefox OS de la fondation Mozilla.

	Ios	BlackBerry	Windows Phone	Android
Langage de programmation	Objective-C	Java	C, C++	Java
	Intégré à Xcode	Gratuit	Gratuit	Gratuit
Disponibilité de l'environnement de développement	Xcode	JDE	Visual Studio, eMbedded VC++	Eclipse, Netbeans, Android Studio
Multiplateforme de déploiement	iPhone, iPod touch, iPad	BlackBerry seulement	Windows Mobile, Windows CE	Android seulement
Cout d'outils de développement	Gratuit	Gratuit	Gratuit	Gratuit
Magasin enLigne	App Store	App World	Windows Market Place	AndroidMarket
Open source	Non	Oui	Non	Oui
Constructeur	Apple	RIM	Microsoft	Google

Tableau 1.1 : Une comparaison entre les systèmes d'exploitation mobile.

III. Le système d'exploitation Android :

Nous avons décrit en grosso modo les différents systèmes d'exploitation utilisés par les applications mobiles. Dans cette section, nous allons mettre le point sur la plateforme qu'on a choisi pour développer notre application. Nous avons choisi le système d'exploitation Android pour les raisons suivantes :

- Android est une plateforme Open Source, puissante, moderne, sûre et ouverte. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de modifier les composants existants. Les utilisateurs peuvent adapter les applications à leur besoin.
- Android est basé sur le noyau Linux. Alors, y a plusieurs avantages comme une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien de bibliothèque partage, etc.
- Le SDK de l'Android offre complètement les APIs pour développer l'application sur Android.
- Grâce au « Android Développeur Challenge » que Google fournira 10 millions de dollars en prix, aucune condition pour les applications sur la plateforme Android, les développeurs ont eu l'opportunité de gagner beaucoup d'argent.








1. La naissance d'Android :






Si on vous dit « Android », une grande majorité d'entre vous me diront « Google ». Vous avez raison, à l'heure actuelle, mais l'histoire commence bien avant l'arrivée de la firme de Mountain View (Californie) sur le système d'exploitation. En effet, tout a débuté avec une société américaine du nom d'Android justement, fondée en 2003.

Celle-ci a été ensuite rachetée par Google deux ans plus tard (en 2005). L'objectif premier était de développer un système d'exploitation qui permettrait à l'utilisateur d'interagir avec ce dernier. Vous le savez peut-être déjà, mais par le passé, chaque constructeur développait son propre système embarqué. Dès lors, il était impossible de concevoir une application compatible sur tous les appareils, sans même parler des bibliothèques de développement fournies qui s'en trouvaient bridées afin que les secrets de fabrication des marques ne soient pas divulgués.

2. L'historique des versions d'Android, une véritable évolution :

Le système de Google n'aurait pas connu un tel succès s'il était resté le même en six ans. C'est là que l'on voit la puissance d'un tel OS qui a su s'adapter aux besoins des utilisateurs à chaque version majeure et qui s'enrichit de nouveautés.

Nom de code	Version	Caractéristiques majeures ajoutées	
Apple pie	1.0	Version connue uniquement ou presque des développeurs car c'est la version du SDK distribuée avant la sortie du premier téléphone Android	
Bananas split	1.1	Ou Beta, version incluse dans le premier téléphone, le HTC G1/Dream	  
Cupcake	1.5	Nouvelles fonctionnalités et mises à jour de l'interface graphique	
Donut	1.6	Nouvelles fonctionnalités et mises à jour de l'interface graphique	
Eclair	2.0	Nouvelles fonctionnalités et mises à jour de l'interface graphique	  
Froyo	2.2.x	Vitesse améliorée, nouvelles fonctionnalités et mises à jour de l'interface graphique	
Gingerbread	2.3.x	Dernière version dédiée uniquement aux smartphones. Cette version est parfois utilisée sur de petites tablettes.	

<p><i>Honeycomb</i></p>	<p>3.x.x</p>	<p>Réservé aux tablettes tactiles et aux téléviseurs connectés¹⁶, cette mise à jour comprend de nombreux changements dans l'interface</p>	
<p>Ice Cream Sandwich</p> <p>Jelly Bean</p> <p>Jelly Bean</p> <p>Jelly Bean</p> <p>KitKat</p>	<p>4.0.x</p> <p>4.1.x</p> <p>4.2.x</p> <p>4.3.x</p> <p>4.4.x</p>	<p>Cette nouvelle version, fortement inspirée d'Honeycomb, unifiée¹⁸ pour smartphones, tablettes et Google TV apporte de nombreux changements</p> <p>Il ajoute un système de notification améliorée, la reconnaissance vocale sans connexion internet, et le « Project Butter » qui augmente la fluidité d'Android.</p> <p>Nouvelle interface de l'appareil photo, d'un système multi-compte uniquement sur tablette, et de Type Gesture permettant d'écrire avec le clavier rien qu'en glissant le doigt.</p> <p>Support du Bluetooth SMART (en) basse consommation et ajout de la norme AVRCP (en) 1.3, gestion multi-utilisateur plus poussée, support d'OpenGL ES 3.0, nouvelle interface de l'appareil photo, mises à jour de sécurité et SlimPort (en).</p> <p>Consommation en ressource moins élevée nécessitant moins de RAM, nouvelles icônes plus soignées, la barre du bas et celle de statut deviennent transparentes sur certains menus et changent de couleur en fonction du contenu affiché.</p>	 <p>4.0</p>  <p>4.1, 4.2, 4.3</p>  <p>4.4</p>
<p>Lollipop</p>	<p>5.0</p>	<ul style="list-style-type: none"> • Nouvelle interface / design ("Material design") • Amélioration de la rapidité • Amélioration de la gestion de la batterie 	




Marshmallow	6.0	<ul style="list-style-type: none"> • Support de l'USB Type-C. • Support de l'authentification par empreinte digitale. • Amélioration de la durée de la batterie avec un mode "deep sleep". • Panneau pour contrôler les permissions des applications. • Android Pay. • Améliorations de Google Now. 	
Nougat	7.0	<ul style="list-style-type: none"> • Meilleur support du multitâche. • Multifenêtrage. • Mises à jour systèmes amélioré (grâce à une double partition système). • Amélioration des performances et de la taille du code grâce à un nouveau compilateur JIT. 	
Nougat	7.1	<ul style="list-style-type: none"> • Eclairage mode nuit. • Amélioration du gestionnaire de stockage. • Amélioration des performances liées à la gestion de l'écran et tactile. • Option pour activer les gestes sur le capteur d'empreinte. • Mises à jour systèmes "transparentes". 	

Tableau 1.2 : Les versions de l'Android.

La **Figure 1.6** nous donne les dernières statistiques qui datent du début février 2017 concernant la répartition des différentes versions Android.

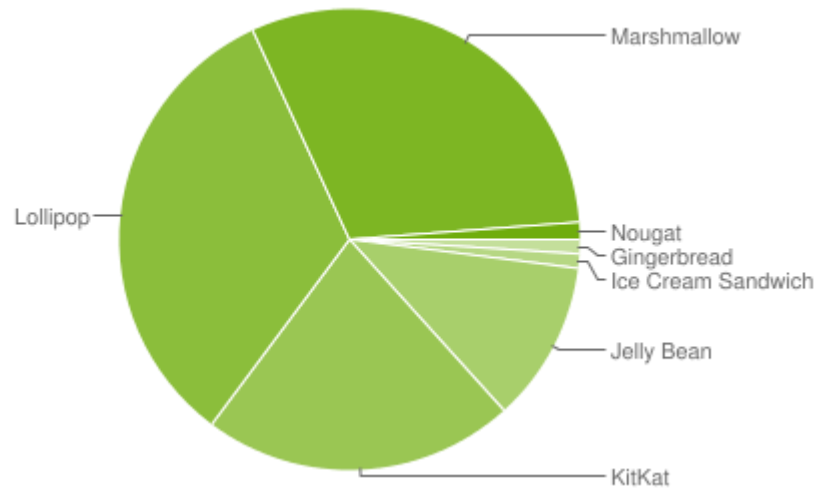


Figure 1.6 : Situation d'Android début février 2017

3. Architecture logicielle :

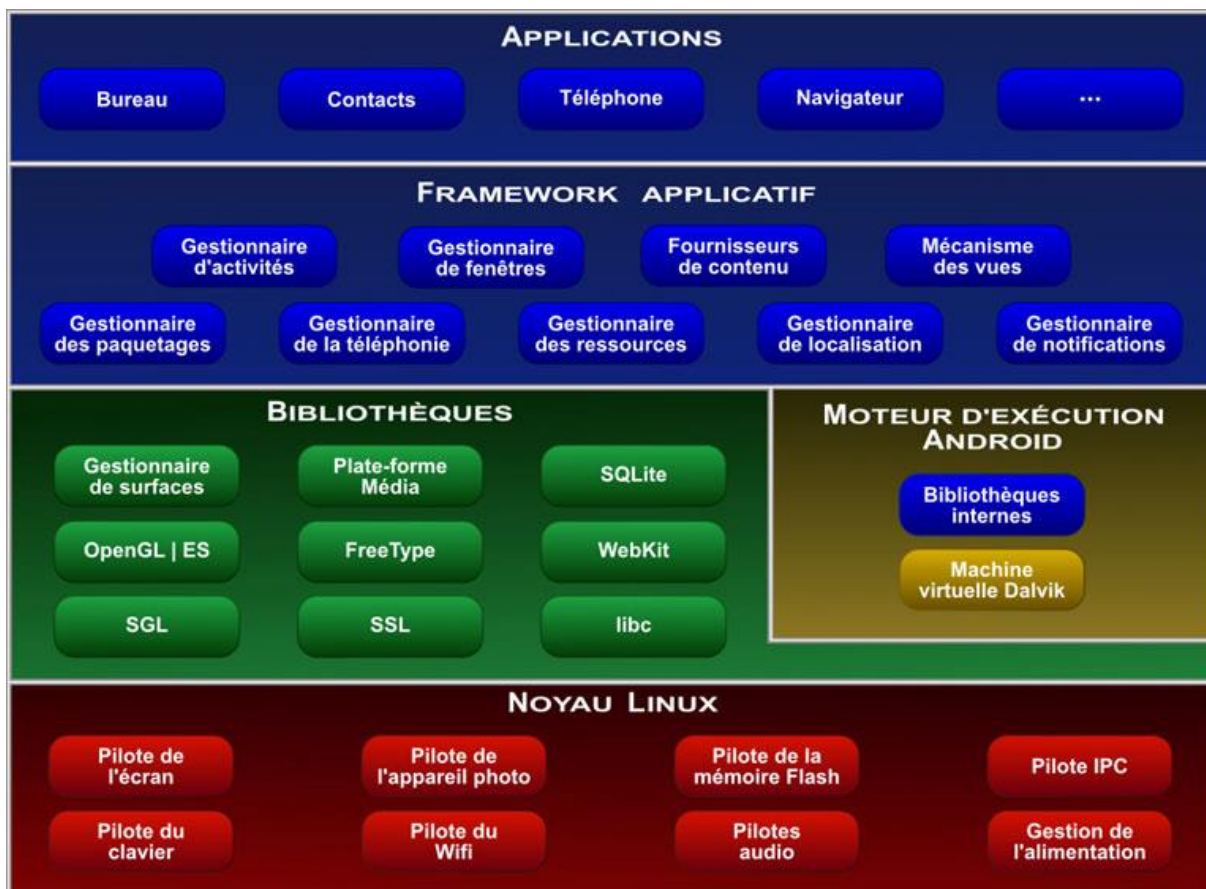


Figure 1.7 : Architecture logicielle de l'Android.

Ainsi, les principaux composants de ce système sont :

a) Le noyau Linux :

Android repose sur un noyau Linux (version 2.6) qui gère les services du système, comme la sécurité, la gestion de la mémoire et des processus, la pile réseau et les pilotes. Il agit également comme une couche d'abstraction entre le matériel et la pile logicielle.

b) Les bibliothèques :

En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android.

c) Moteur d'exécution Android :

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de machine virtuelle Dalvik.

d) Le Framework :

Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes.

e) Applications :

Android est fourni avec un ensemble d'applications natives permettant d'accéder à des fonctionnalités comme les courriels, le téléphone, le calendrier, les photos... Ces applications sont développées à l'aide du langage de programmation Java.

Pour l'utilisateur final, c'est la seule couche accessible et visible.

4. Kit de développement :

Exploiter une nouvelle plate-forme n'est jamais été une chose aisée. C'est pourquoi Google fournit, en plus du système d'exploitation, un kit de développement (Software Development Toolkit ou SDK).

Ce SDK est un ensemble d'outils qui permet aux développeurs et aux entreprises de créer des applications. Il est disponible gratuitement sur le site de Google.

Le SDK Android est composé de plusieurs éléments afin d'aider les développeurs à créer et à maintenir des applications :

- ✓ Des outils.
- ✓ Des exemples de code.
- ✓ De la documentation.
- ✓ Des API (interfaces de programme d'application).

a) Les outils :

Le SDK est livré avec un certain nombre d'outils couvrant différents aspects du cycle de développement d'une application Android. Le kit de développement propose une boîte à outils complète pour les tâches de compilation, de débogage, de génération de code AIDL et de la signature de l'application, etc.

L'émulateur Android : c'est un téléphone virtuel qui permet de tester les applications qui sont en train de se développer. Il est lancé par la commande "emulator ". Celle-ci prend en paramètre l'image AVD (Android Virtual Device) qui sera montée en mémoire. Il a des limitations par exemple : il n'est pas capable de supporter le Bluetooth ainsi qu'il ne permet pas le teste des applications de réalité augmentée.

b) Les exemples de code :

Le kit de développement est accompagné d'un certain nombre d'exemples illustrant les possibilités du SDK Android. Parmi ces exemples, on peut citer : un jeu du serpent et le projet qui couvre l'utilisation de plusieurs exemples de l'API Android comme les alarmes, les notifications et les menus.

c) La documentation :

La documentation du SDK Android est scindée en deux parties bien distinctes :

- ✓ Le guide du développeur qui est disponible en HTML (Hypertext Markup Language) dans le répertoire du SDK qu'on vient d'installer ;
- ✓ La documentation des API au format java doc est également située dans le répertoire docs et accessible grâce au chemin débutant du répertoire d'installation.

d) Les API :

Android offre plusieurs API (Application Program Interface) tel que :

- ✓ Google Maps : intègre et contrôle l'affichage d'une carte dans une interface graphique de l'application.
- ✓ Géolocalisation : permet d'accéder au service de localisation du système, de choisir le fournisseur en fonction des critères et de préciser la position actuelle du téléphone (latitude, longitude, vitesse, etc.).

IV. La Géolocalisation :

Parmi les fonctionnalités les plus appréciées sur les plates-formes mobiles modernes, la géolocalisation permet de réaliser des applications innovantes. Grâce à Google Maps notamment, elle est au cœur d'Android. Le service de géolocalisation récupère les coordonnées de l'utilisateur et les envoie pour la réalisation un service informatique. A chaque fois, il demande l'autorisation de l'utilisateur avant la réalisation de cette opération. Seules les informations concernant la latitude et la longitude sont envoyées.

La localisation du mobile se fait selon plusieurs technologies comme :

- a) **GPS** : Il s'effectue par la réception de signaux provenant de plusieurs satellites qui se trouve en orbite. Le téléphone mobile équipé d'un GPS permettra de transmettre sa position via un réseau SMS, GPRS, Edge ou UMTS.
- b) **Internet** : La précision de la localisation par adresse IP sur le réseau internet se situera au niveau d'un pays, d'une ville ou d'un quartier selon l'opérateur (national ou local). Cependant, au sein d'un réseau ADSL d'un même opérateur, la géolocalisation peut être très

précise (adresse ou bâtiment par exemple) si les lieux des connexions sont enregistrés dans une base de données.

c) **Wifi** : La localisation est similaire au cas du réseau GSM ou IP, par les cellules émettrices, avec une précision inférieure à 100 mètres. Une triangulation entre plusieurs antennes Wifi peut donner la position avec une précision d'environ 5m par l'analyse de la puissance du signal radio reçu de l'appareil.

d) **GSM** : Il est basé sur le code unique de la carte SIM. La connexion au réseau est autorisée après une identification à une cellule composant le réseau GSM. La précision dépend de l'étendue de la cellule, de 250 mètres en zone urbaine à 10 km en zone rurale.

1. Google Maps :

Google Maps est un service gratuit de carte géographique et de plan en ligne. Le service a été créé par Google. Il s'agit d'une forme de géoportail. Lancé en 2004 aux États-Unis et au Canada et en 2005 en Grande Bretagne (sous le nom de Google Local), Google Maps a été lancé jeudi 27 avril 2006, simultanément en France, Allemagne, Espagne et Italie.

Ce service a ceci de particulier qu'il permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue. Deux types de plan sont disponibles : un plan classique, avec nom des rues, quartier, villes et un plan en image satellite, qui couvre aujourd'hui le monde entier.

C'est au début du mois d'avril 2005 que Google Maps s'enrichit de la vue par image satellite, en plus de la cartographie classique. Se différenciant des autres services proposant des images satellites statiques, Google Maps permet de naviguer et de se positionner où l'on veut sur la carte satellite.

Dans notre application nous utilisons Google Map pour chercher les itinéraires des médecins de Tlemcen et lancer la navigation GPS

V. Les bases de données :

Les bases de données jouent un rôle sans cesse croissant dans les systèmes d'information d'entreprise, qu'il s'agisse d'applications de gestion traditionnelles (comptabilité, ventes, décisionnel...) ou d'applications intranet e-commerce ou de gestion de la relation client. Comprendre les principes des bases de données, les langages d'interrogation et de mise à jour, les techniques d'optimisation et de contrôle des requêtes, les méthodes de conception et la gestion des transactions devient une nécessité pour tous les professionnels et futurs professionnels de l'informatique.

1. Qu'est-ce qu'une base de données ?

Une base de données, ce mot est souvent utilisé pour désigner n'importe quel ensemble de données, il s'agit là d'un abus de langage qu'il faut éviter. Une base de données est un ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique. Pour mériter le terme de base de données, un ensemble de données non indépendantes doit être interrogeable par le contenu, c'est-à-dire que l'on doit

pouvoir retrouver tous les objets qui satisfont à un certain critère, par exemple tous les produits qui coûtent moins de 100francs. Les données doivent être interrogeables selon n'importe quel critère. Il doit être possible aussi de retrouver leur structure, par exemple le fait qu'un produit possède un nom, un prix et une quantité.

Les bases de données ont pris aujourd'hui une place essentielle dans l'informatique, plus particulièrement en gestion. Au cours des trente dernières années, des concepts, méthodes et algorithmes ont été développés pour gérer des données sur mémoires secondaires ; ils constituent aujourd'hui l'essentiel de la discipline «Bases de Données» (BD). Cette discipline est utilisée dans de nombreuses applications. Il existe un grand nombre de Systèmes de Gestion de Bases de Données (SGBD) qui permettent de gérer efficacement de grandes bases de données. De plus, une théorie fondamentale sur les techniques de modélisation des données et les algorithmes de traitement a vu le jour. Les bases de données constituent donc une discipline s'appuyant sur une théorie solide et offrant de nombreux débouchés pratiques.

Utilisation de moyens informatiques pour répondre à un besoin déterminé en faisant appel de manière importante à une ou plusieurs bases de données à travers un système de gestion de bases de données (SGBD) (Database application).

Il existe plusieurs types de BD :

a) Bases de données hiérarchiques :

Historiquement, le premier système de base de données a été conçu pour la gestion des données du programme Apollo de la NASA. Les données étaient structurées dans des hiérarchies, comparables à l'organisation des répertoires sur un PC. Une base de données hiérarchique est une base de données dont le système de gestion lie les enregistrements dans une structure arborescente où chaque enregistrement n'a qu'un seul possesseur.

Les structures de données hiérarchiques ont été largement utilisées dans les premiers systèmes de gestion de base de données. Elles ont toutefois montré des limites pour décrire des structures complexes, répondre aux besoins réels et suivre l'évolution des systèmes d'information.

b) Bases de données réseaux :

Ce modèle de base de données a été inventé par Charles.W. Bachmann pour lequel il reçut en 1973 le prix Turing. Le modèle réseau est une manière de représenter les données dans le cadre d'une base de données. Ce modèle est en mesure de lever de nombreuses difficultés du modèle hiérarchique grâce à la possibilité d'établir des liaisons de type 1-N en définissant des associations entre tous les types d'enregistrements. Ce modèle est une extension du modèle précédent (hiérarchique), les liens entre objets peuvent exister sans restriction. Pour retrouver une donnée dans une telle modélisation, il faut connaître son chemin d'accès (les liens). Ceci rend cependant les programmes encore dépendants de la structure de données.

c) Bases de données relationnelles :

En 1970, au moment où les systèmes basés sur le modèle hiérarchique ou le modèle en réseau étaient en plein développement, CODD proposa de stocker des données hétérogènes dans des tables, permettant d'établir des relations entre elles. De nos jours, ce modèle est extrêmement répandu mais en 1970, cette idée était considérée comme une curiosité intellectuelle. On doutait alors que des tables puissent être jamais gérées de manière efficace par un ordinateur. Ce scepticisme n'a cependant pas empêché CODD de poursuivre ses recherches, et IBM construira rapidement un premier prototype de Système de Gestion de Base de Données Relationnelles (SGBDR).

Ce mode d'organisation des données a donné naissance à une pléthore de SGBD relationnels commerciaux et non commerciaux dont DB2 d'IBM, Oracle10, Microsoft Access, MySQL pour ne nommer que les mieux connus. Les SGBD relationnels sont de deuxième génération et ils ont tous en commun intrinsèquement un langage appelé SQL (Structured Query Language) SQL agissant à la fois comme langage de définition et langage de manipulation de données.

2. Qu'est-ce qu'un SGBD ? (Data Base Management System)

Un SGBD peut apparaître comme un outil informatique permettant la sauvegarde, l'interrogation, la recherche et la mise en forme de données stockées sur mémoires secondaires. Ce sont là les fonctions premières, complétées par des fonctions souvent plus complexes, destinées par exemple à assurer le partage des données mais aussi à protéger les données contre tout incident et à obtenir des performances acceptables. Les SGBD se distinguent clairement des systèmes de fichiers par le fait qu'ils permettent la description des données (définition des types par des noms, des formats, des caractéristiques et parfois des opérations) de manière séparée de leur utilisation (mise à jour et recherche). Ils permettent aussi de retrouver les caractéristiques d'un type de données à partir de son nom (par exemple, comment est décrit un produit). Le système de fichiers est un composant de plus bas niveau ne prenant pas en compte la structure des données. La tendance est aujourd'hui à intégrer le système de fichiers dans le SGBD, construit au-dessus.

En conséquence, un SGBD se compose en première approximation de trois couches emboîtées de fonctions, depuis les mémoires secondaires vers les utilisateurs (**figure 1.8**) :

✓ La gestion des récipients de données sur les mémoires secondaires constitue traditionnellement la première couche ; c'est le gestionnaire de fichiers, encore appelé système de gestion de fichiers. Celui-ci fournit aux couches supérieures des mémoires secondaires idéales adressables par objets et capables de recherches par le contenu des objets (mécanismes d'indexation notamment).

✓ La gestion des données stockées dans les fichiers, l'assemblage de ces données en objets, le placement de ces objets dans les fichiers, la gestion des liens entre objets et des structures permettant d'accélérer les accès aux objets constituent la deuxième couche ; c'est le système d'accès aux données ou SGBD interne. Celui-ci repose généralement sur un modèle de données internes, par exemple des tables reliées par des pointeurs.

✓ La fonction essentielle de la troisième couche consiste dans la mise en forme et la présentation des données aux programmes d'applications et aux utilisateurs interactifs. Ceux-ci expriment leurs critères de recherches à l'aide de langages basés sur des procédures de recherche progressives ou sur des assertions de logiques, en référençant des données dérivées de la base; c'est le SGBD externe qui assure d'une part l'analyse et l'interprétation des requêtes utilisateurs en primitives internes, d'autre part la transformation des données extraites de la base en données échangées avec le monde extérieur.

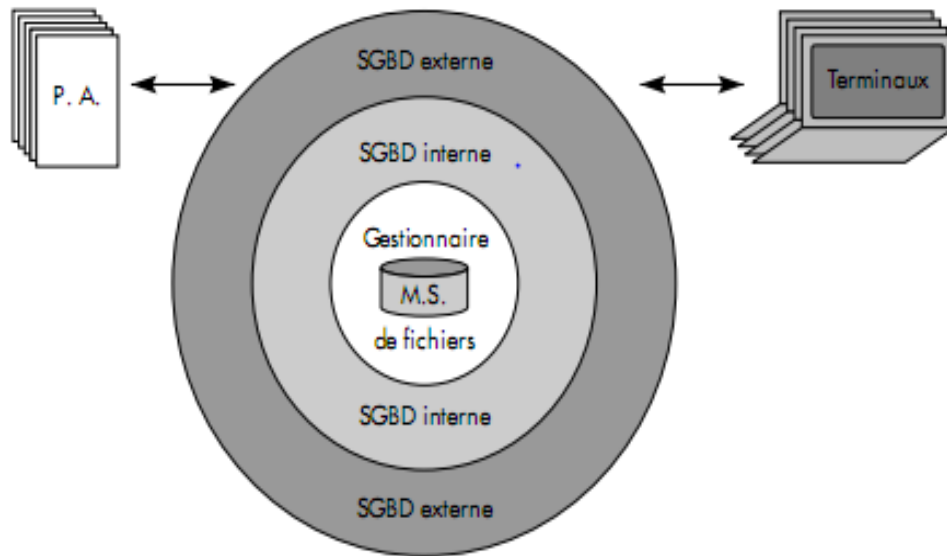


Figure 1.8 : Première vue d'un SGBD.

P.A. = Programmes d'Application

M.S. = Mémoires Secondaires

Les données stockées dans des bases de données modélisent des objets du monde réel, ou des associations entre objets. Les objets sont en général représentés par des articles de fichiers, alors que les associations correspondent naturellement à des liens entre articles. Les données peuvent donc être vues comme un ensemble de fichiers reliés par des pointeurs ; elles sont interrogées et mises à jour par des programmes d'applications écrits par les utilisateurs ou par des programmes utilitaires fournis avec le SGBD (logiciels d'interrogation interactifs, éditeurs de rapports, etc.). Les programmes sont écrits dans un langage de programmation traditionnel appelé langage de 3e génération (C, COBOL, FORTRAN, etc.) ou dans un langage plus avancé intégrant des facilités de gestion d'écrans et d'édition de rapports appelé langage de 4e génération (Visual BASIC, SQL/FORMS, MANTIS, etc.). Dans tous les cas, ils accèdent à la base à l'aide d'un langage unifié de description et manipulation de données permettant les recherches et les mises à jour (par exemple, le langage SQL). Cette vision simplifiée d'un SGBD et de son environnement est représentée (**Figure 1.9**) :

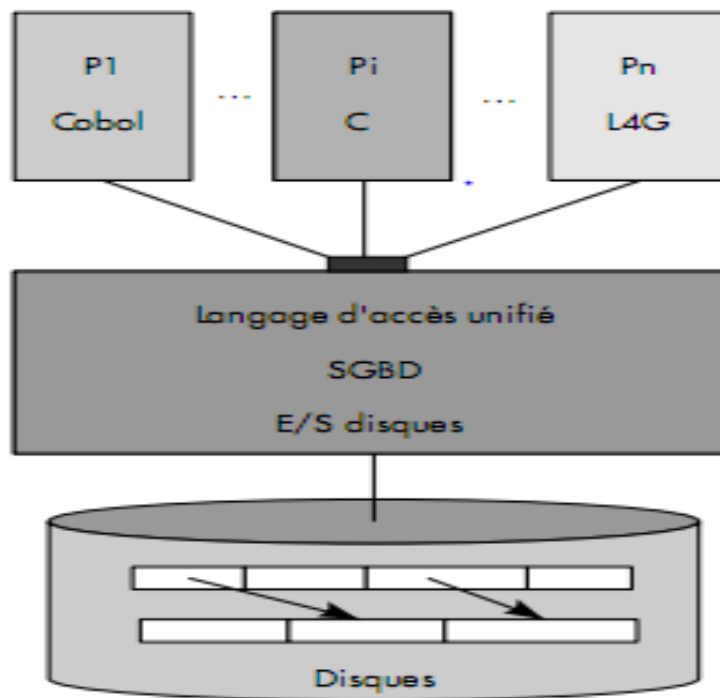


Figure 1.9 : Composants d'un environnement base de données.

a) Les premiers objectifs d'un SGBD :

- Indépendance physique des programmes aux données
- Indépendance logique des programmes aux données
- Manipulation des données par des langages non procéduraux
- Administration facilitée des données.

Voici les objectifs additionnels des SGBD, qui sont en fait des conséquences des objectifs premiers :

- Efficacité des accès aux données
- Partage des données
- Cohérence des données
- Redondance contrôlée des données
- Sécurité des données.

A l'heure actuelle, les SGBDR sont présents dans de nombreux logiciels, sont très répandus dans les bases de données et représentent une industrie de plusieurs milliards de dollars.

b) Principe de fonctionnement d'un SGBDR :

La BD relationnelle permet de stocker et d'organiser une grande quantité d'information. Les SGBDR permettent de naviguer dans ces données et d'extraire (ou de mettre à jour) les informations voulues au moyen d'une requête. Les données apparaissent comme stockées dans des tables qu'on nomme également relations.

Ce modèle relationnel conduit à :

- une grande simplicité d'usage
- une transparence pour l'utilisateur de toute réorganisation technique de la base
- une facilité de combinaison du contenu de plusieurs tables (opération join ou jointure).

Les relations possèdent un certain nombre d'attributs permettant de décrire un enregistrement. La non-duplication (absence de redondance) des enregistrements est assurée par le SGBDR. Dans les relations, il est possible de définir deux types de clés :

- une clé primaire permet d'identifier un et un seul enregistrement (par exemple le numéro de sécurité sociale). Deux enregistrements ne peuvent donc avoir la même clé primaire.
- une clé étrangère, attribut d'une relation qui est une clé primaire dans une autre table. Elle permet donc de lier deux relations entre elles.

Pour relier les données, les jointures sont utilisées, dont il existe différents types : un-à-un, un-à plusieurs, ...

Lors d'une recherche de données, les opérations sont communiquées sous forme de requêtes aux SGBDR. Une requête se présente sous la forme d'une table, que l'opérateur remplit en indiquant pour quel champ il désire réaliser l'interrogation, et avec quel(s) critère(s). Comme nous l'avons dit plus haut, le SGBDR s'occupe de traduire cette demande en langage SQL.

Dans une base de données relationnelle, le but est de séparer les informations au maximum pour éviter les doublons et la redondance, et d'empêcher la perte de qualité d'information (par exemple, l'adresse d'un fournisseur n'est mise à jour qu'une et une seule fois : la modification sera alors prise en compte sur l'ensemble des courriers).

Le langage unifié de description et manipulation de données : **SQL (Structured Query Language)**

C'est un langage standard de requêtes qui permet d'interroger les bases de données de manière simple. Le SQL est facilement compréhensible par l'humain et par les différents SGBDR. D'une manière générale, les instructions pour l'interrogation de la base constituent ce que l'on appelle une requête SQL. Une requête SQL pourrait être alors une modification, une insertion, une suppression, ou une sélection de données, voire même une création de table.

Le SQL se décompose en 5 parties :

- Ordres DDL (Data Definition Language) : permettant de modifier la structure de la base de données
- Ordres DML (Data Manipulation Language) : permettant de consulter / modifier le contenu de la base de donnée
- Ordres DCL (Data Control Language) : permettant de gérer les privilèges, c'est-à-dire les utilisateurs et les actions qu'ils peuvent entreprendre.
- Ordres TCL (Transaction Control Language) : permettant de gérer les transactions, c'est-à-dire rendre atomique divers ordres enchaînés en séquence
- SQL procedural: PSM (Persistent Stored Module), CLI (Call Level Interface), Embedded SQL, Qui est un ensemble d'outils pour que SQL s'interface avec des langages hôtes.

c) Le SGBDR choisit :

Lors du développement de notre projet, la question s'est posée du choix du SGBDR. Il nous a semblé opportun d'opter pour un logiciel largement diffusé, possédant une notice technique importante ainsi qu'une grande communauté de développeurs, afin de pouvoir en tirer aide et conseils durant le cheminement de l'élaboration du projet.

MySQL est un Système de Gestion de Bases de Données Relationnelles, qui utilise le langage SQL. C'est un des SGBDR le plus utilisés. Sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel Open Source, ce qui signifie que son code source est librement disponible et que quiconque qui en ressent l'envie et/ou le besoin peut modifier MySQL pour l'améliorer ou l'adapter à ses besoins. Une version gratuite de MySQL est par conséquent disponible. À noter qu'une version commerciale payante existe également.

Le développement de MySQL commence en 1994 par David Axmark et Michael Widenius. EN 1995, la société MySQL AB est fondée par ces deux développeurs, et Allan Larsson. C'est la même année que sort la première version officielle de MySQL.

En 2008, MySQL AB est rachetée par la société Sun Microsystems, qui est elle-même rachetée par Oracle Corporation en 2010.

On craint alors la fin de la gratuité de MySQL, étant donné qu'Oracle Corporation édite un des grands concurrents de MySQL : Oracle Data base, qui est payant (et très cher). Oracle a cependant promis de continuer à développer MySQL et de conserver la double licence GPL (libre) et commerciale jusqu'en 2015 au moins.

MySQL consiste en un ensemble de programmes chargés de gérer une ou plusieurs bases de données, et qui fonctionnent selon une architecture client/serveur (**Figure 1.10**).

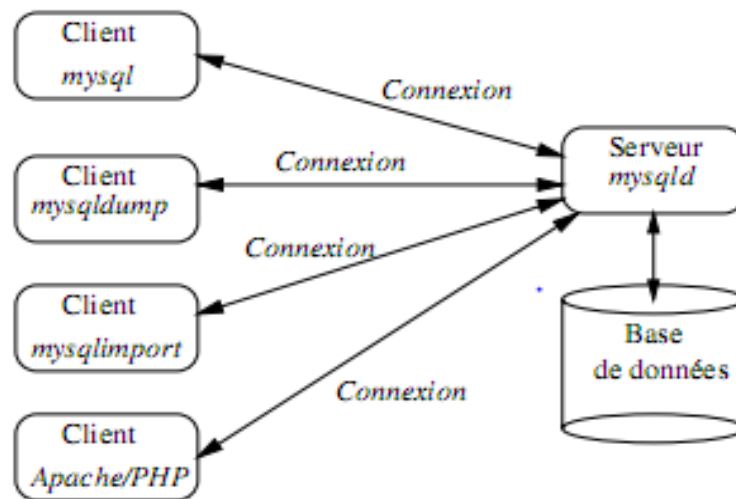


Figure 1.10 : Serveur et clients de MySQL.

Le serveur `mysqld`. Le processus `mysqld` est le serveur de MySQL. Lui seul peut accéder aux fichiers stockant les données pour lire et écrire des informations.

Utilitaires. MySQL fournit tout un ensemble de programmes, que nous appellerons utilitaires par la suite, chargés de dialoguer avec `mysqld`, par l'intermédiaire d'une connexion, pour accomplir un type de tâche particulier. Par exemple, `mysqldump` permet d'effectuer des sauvegardes, `mysqlimport` peut importer des fichiers ASCII dans une base, etc.

Le client le plus simple est simplement nommé MySQL, et permet d'envoyer directement des commandes au serveur.

Conclusion :

Dans ce chapitre, nous avons présenté les applications mobiles, les technologies associées et plus spécifiquement le système d'exploitation Android en étalant ces fonctionnalités, son architecture ainsi que les principaux composants du système. Ensuite nous avons présenté une idée générale sur la géo-localisation et les bases de données.

CHAPITRE 2 :
Outils utilisés et conception UML

Introduction :

Après l'émergence des technologies mobiles beaucoup d'applications ont été développées dans divers domaines. Cependant, nous avons pensé à utiliser ces gadgets pour créer une application pour aider chaque individu à trouver un médecin en cas de besoin à partir de son Smartphone.

Ce type d'application métier s'avère très utile non seulement afin de subvenir aux besoins des patients mais aussi pour les avantages qui présente pour les médecins. Il s'agit de rapprocher le médecin de ses patients et de le rendre plus disponible et surtout facilement accessible en cas de besoin.

Dans ce chapitre nous présentons notre application en commençant par la description de notre problématique suivie de l'environnement de travail utilisé ainsi que les besoins fonctionnels et non fonctionnels.

I. Problématique :

Le marché des applications mobiles connaît une croissance exponentielle, tiré par une demande des usagers, tant professionnels que grand public, et accompagné par les progrès technologiques et l'évolution des terminaux. Cela ouvre de nouvelles opportunités aux développeurs d'applications mobiles pour proposer et mettre en œuvre leurs produits dans tous les domaines et sans être limités Le commerce, La médecine, L'éducation, L'économie etc.

De nos jours avec la multiplicité des médecins et des spécialistes, le patient a des difficultés pour aller au médecin qu'il voulut soit le plus proche ou le plus connais. En plus il besoin de connaître l'itinéraire vers lui, donc la seule méthode qu'il peut appliquer est de demander l'aide à des gens plus expérimentés. Ce problème se trouve surtout dans les grandes villes.

Dans ce travail, nous proposons une solution pour faciliter la recherche aux patients. Nous développons une application 'DOCTOR FINDER' pour les Smartphones et les tablettes tactiles sous Android qui aura vocation à être un portail d'information pour les clients afin de chercher et situer le médecin le plus proche. L'application utilise une Google Map pour pouvoir voir la localisation de ces médecins, ainsi d'avoir la possibilité de les contacter par appel téléphonique.

Ce travail est justifié par la disponibilité des appareils intelligents (téléphone sous Android), ainsi que l'assurance, la rigueur et la vitesse que nous garantit la technologie.

Ce travaille simplifiera la façon avec laquelle est cherchée géographiquement et visualisée très rapidement les médecins existant dans la région.

II. Environnement de travail :

1. SDK :

SDK signifie Software Development Kit, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc. pour un terminal et/ou un système d'exploitation spécifique.

Depuis l'arrivée de Android, le système d'exploitation que Google a développé pour se lancer dans le secteur mobile, sont un kit de développement est désormais disponible.

Le kit de développement Android inclut les utilitaires nécessaires pour aider les développeurs Android avec les premières étapes: les différents API développés par Google à la fois pour contrôler les fonctions de l'appareil et l'intégration des services, un émulateur complet pour tester des applications, et tout le matériel de lecture nécessaire pour vous aider à effectuer les premières étapes de la programmation pour Android.

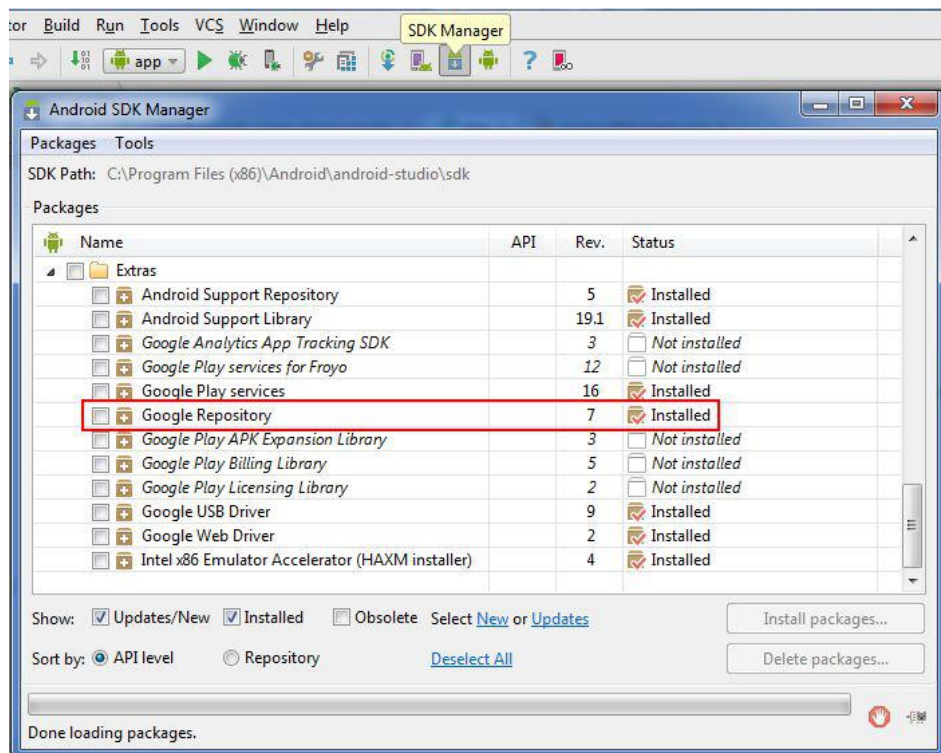


Figure 2.1 : SDK Manager.

2. Android Studio :

Depuis décembre 2014, l'environnement de développement Eclipse, avec son module ADT ont pris leur retraite afin de faire place au tout nouvel environnement Android Studio.

Android Studio représente la plateforme officielle, soutenue par Google, pour le développement d'applications Android. Il repose sur IntelliJ (Community Edition) de JetBrains et devrait permettre aux développeurs d'être plus rapides et plus productifs. Il offre toutes les possibilités nécessaires pour développer une application Android complète.

Android Studio est annoncé le 15 mai 2013 lors du Google I/O et une version "Early Access Preview" est disponible le jour même.

Le 8 décembre 2014, Android Studio passe de la version bêta à la version stable 1.0. Android Studio remplace alors Eclipse ADT en tant qu'IDE officiel.



Figure 2.2 : Logo de la plateforme Android Studio.

Le 7 avril 2016, la version v2.0 est centrée sur la performance. Elle introduit un émulateur beaucoup plus rapide et depuis le 19 septembre 2016, c'est la dernière version stable v2.2 Elle supporte Java 1.8, et améliore les performances globales.

Tout projet Android doit respecter une hiérarchie bien précise qui permettra au compilateur de retrouver les différents éléments et ressources lors de la génération de l'application. Cette hiérarchie favorise la modularité des applications Android.

A la création du projet, Android Studio crée automatiquement des dossiers pour contenir les fichiers de code Java, les fichiers XML, et les fichiers multimédias.

L'explorateur de projet vous permettra de naviguer dans ces dossiers.

Les dossiers les plus utilisés sont **java** et **res**. Le premier contient le code Java qui définit le comportement de l'application (situé dans le répertoire de votre projet sous **app\src\main**) et la seconde comporte des sous dossiers (dans **app\src\main\res**) où sont stockées les ressources qui définissent l'interface de l'application (l'apparence).

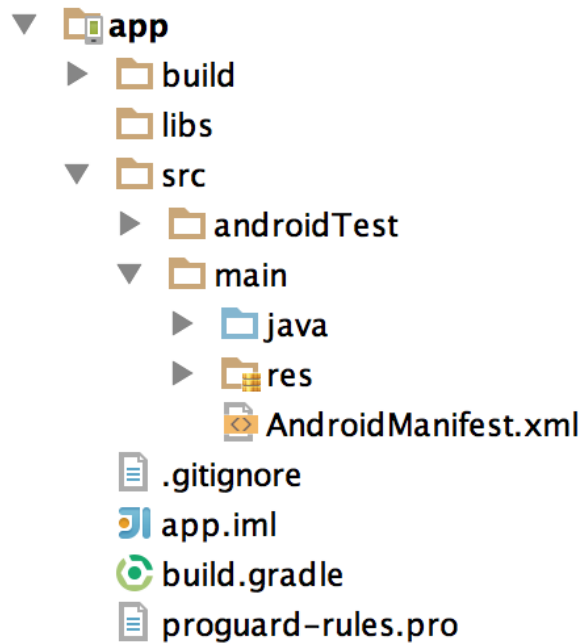


Figure 2.3 : Structure d'un projet Android.

3. Serveurs :

La partie serveur est composé de deux serveurs distants : le serveur web et le serveur de base de donnée.

- Le serveur Web utilisé est le serveur Apache est le serveur le plus répondu sur Internet pour déployer les scripts PHP.
- Le serveur de base de données utilisé est le serveur MySQL pour stocker les données
- Pour créer et géré notre base de donnée nous avons utilisé l'outil phpMyAdmin

PhpMyAdmin :(PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

4. WampServer :

WampServer est une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.



Figure 2.4 : Logo WampServer et ses comprenant.

5. Intégration de Google Map :

Pour utiliser les cartes Google, il faut suivre les étapes suivantes :

- 1) Charger la bibliothèque Google Play Services SDK.

Extras			
<input type="checkbox"/>	Android Support Library	12	Installed
<input type="checkbox"/>	Google AdMob Ads SDK	9	Installed
<input type="checkbox"/>	Google Analytics App Tracking SDK	2	Not installed
<input type="checkbox"/>	Google Analytics SDK	2	Installed
<input type="checkbox"/>	Google Cloud Messaging for Android Library	3	Installed
<input type="checkbox"/>	Google Play services	5	Installed
<input type="checkbox"/>	Google Play APK Expansion Library	3	Installed
<input type="checkbox"/>	Google Play Billing Library	4	Installed
<input type="checkbox"/>	Google Play Licensing Library	2	Installed
<input type="checkbox"/>	Google USB Driver	7	Installed
<input type="checkbox"/>	Google Web Driver	2	Installed
<input type="checkbox"/>	Intel Hardware Accelerated Execution Manager	2	Installed

Figure 2.5 : Google Play Services.

- 2) Obtenir une Maps API v2 Key.

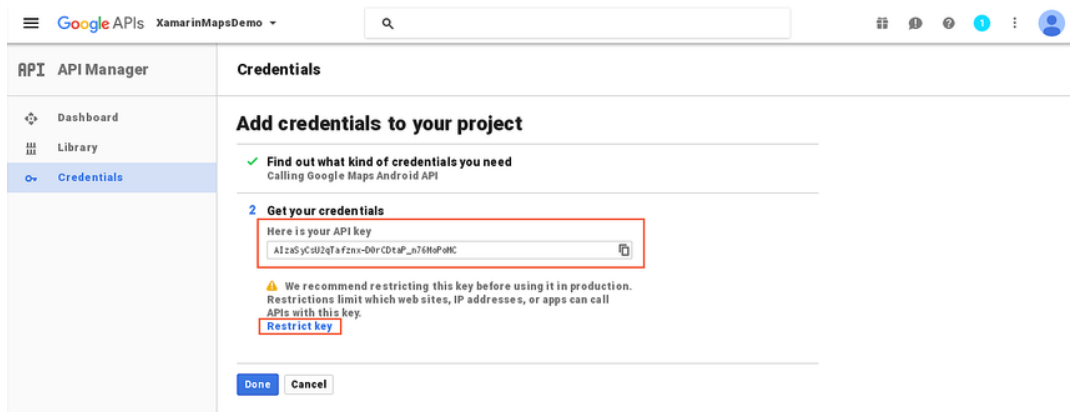


Figure 2.6 : La clé API de Google Map.

- 3) Configurer l'AndroidManifest.xml de l'application :

```

<meta-data
    android:name="com.google.android.gms.version"
    android:value="4323000" />
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyC1aH20Xc0iAlpho_3fkiSicK-z2GrXi94" />

<activity
    android:name=".MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <uses-library android:name="com.google.android.maps" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
</activity>
    
```

Figure 2.7 : La clé API au fichier AndroidManifest.

- 4) Ecrire une activité demandant à afficher une carte Google :

Le code de l'activité est :

```
import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Figure 2.8 : Le code pour MainActivity.java.

Tout est donc dans l'activity_main.xml qui est :

```
<?xml version="1.0" encoding="utf-8"?>
<fragment
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Figure 2.9 : Le code pour l'activity_main.xml.

Toute la procédure est indiquée à :

<https://developers.google.com/maps/documentation/android/start>

6. Photoshop :

Photoshop est un logiciel de traitement et de retouche de photos et d'images produit par Adobe, une compagnie spécialisée en infographie. Il est grandement utilisé par les infographistes grâce aux multiples outils dont il est doté et à la gamme d'effets qu'il permet de produire pour des réalisations des plus originales. Il est compatible avec les autres logiciels de cette compagnie, dont Illustrator, ImageReady, Acrobat, etc.

Il permet de modifier ou de créer des images, de fusionner des images, de réaliser des logos, de corriger des couleurs, d'appliquer de multiples effets et des filtres, de créer des fonds et des puces, d'ajouter des effets spéciaux, de faire des montages, etc. Bref, le mot d'ordre à ne pas oublier avec Photoshop est créativité !



III. Besoins et conception :

Les besoins de l'application se divisent en des besoins non fonctionnels et fonctionnels.

1. Les besoins non fonctionnels :

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d'objectifs spécifiques que doit atteindre le système.

Les besoins non fonctionnels de cette application sont les suivants :

- ✓ L'application doit garantir la sécurité à travers la gestion des droits d'accès.
- ✓ L'application doit être facile à utiliser.
- ✓ L'accès à la base de données doit être souple et rapide.
- ✓ L'application doit être toujours fonctionnelle.
- ✓ La confidentialité des données personnelles (Médecins).

2. Les besoins fonctionnels :

Cette partie est réservée à la description des exigences fonctionnelles des acteurs de l'application :

- ❖ **L'Utilisateur** : quand le patient utilise notre application, il peut :
 - ✓ Chercher un médecin précis et avoir la carte géographique qui localise le médecin en question avec l'itinéraire vers lui.
 - ✓ Chercher les médecins à proximité et avoir la carte géographique qui les localise.
 - ✓ Avoir les informations sur un médecin.
 - ✓ Un accès rapide à des numéros utile en cas d'urgence.

- ❖ **Le médecin** : il fait tout ce que fait un utilisateur simple en plus :
 - ✓ Gérer ses informations : Ajouter, modifier, ou supprimer.

IV. Conception UML de notre application :

Comme n'importe quel type de projet, notre projet nécessite une phase de conception. Pour cela nous avons utilisés les diagrammes en UML.

1. Définition de l'UML :

C'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter.

UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

Ce langage ne préconise aucune démarche, ce n'est donc pas une méthode. Chacun est libre d'utiliser les types de diagramme qu'il souhaite, dans l'ordre qu'il veut. Il suffit que les diagrammes réalisés soient cohérents entre eux, avant de passer à la réalisation du logiciel.

2. Diagramme de cas d'utilisations :

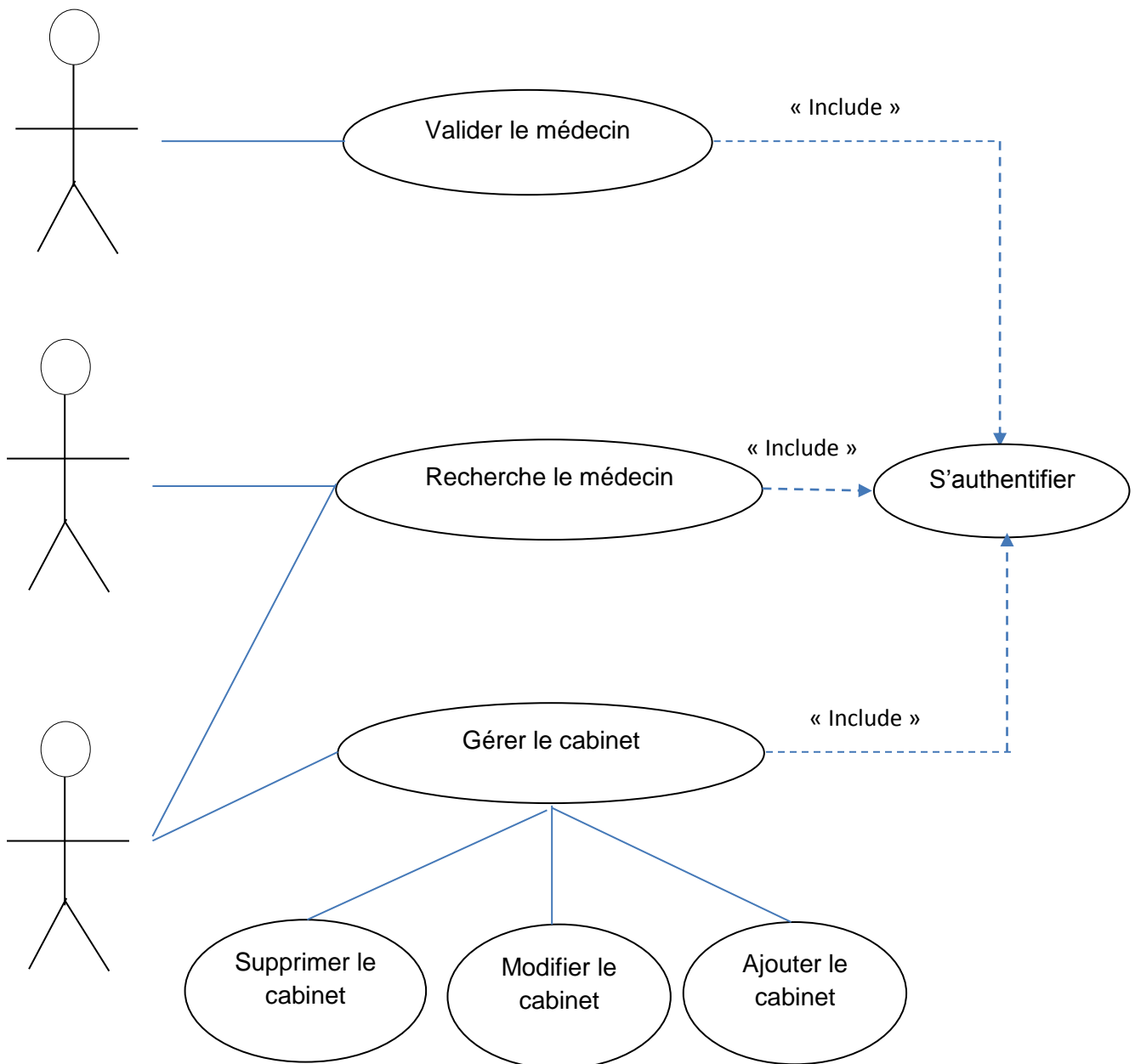


Figure 2.10 : Diagrammes des cas d'utilisations.

3. Diagrammes de classes :

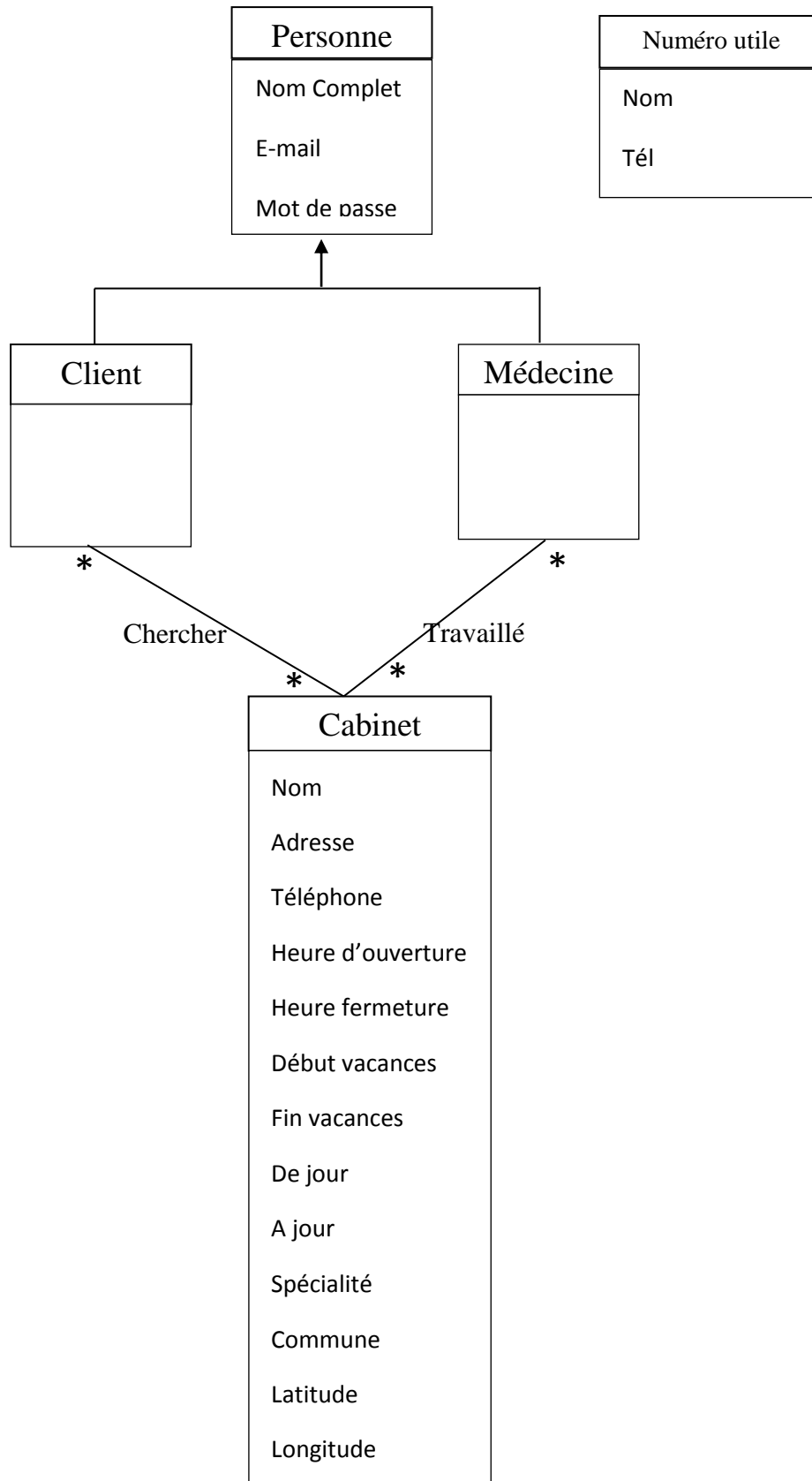


Figure 2.11 : Diagrammes de classe.

4. Diagrammes d'activité :

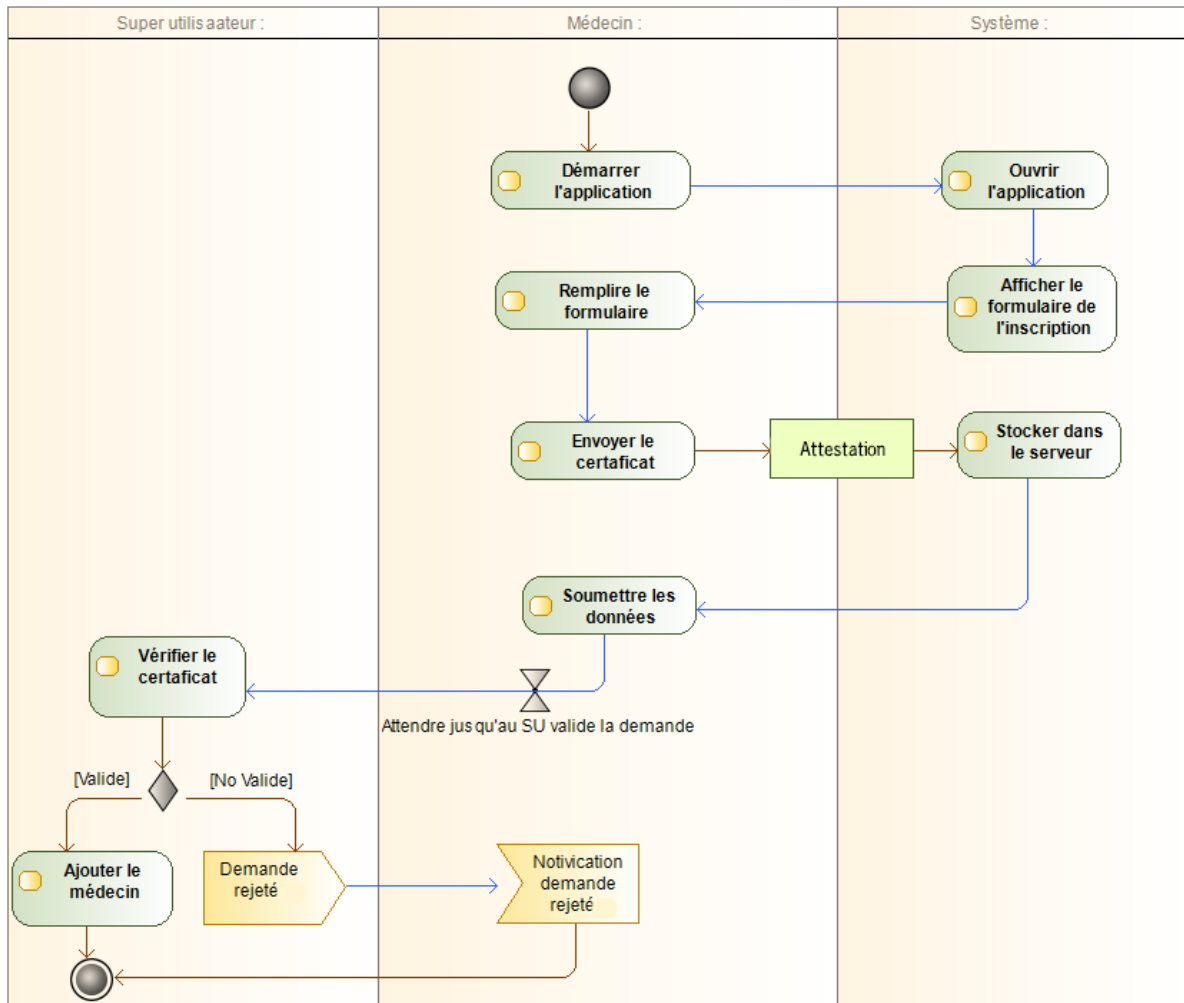


Figure 2.12 : Diagramme d'activité -Ajouter cabinet.

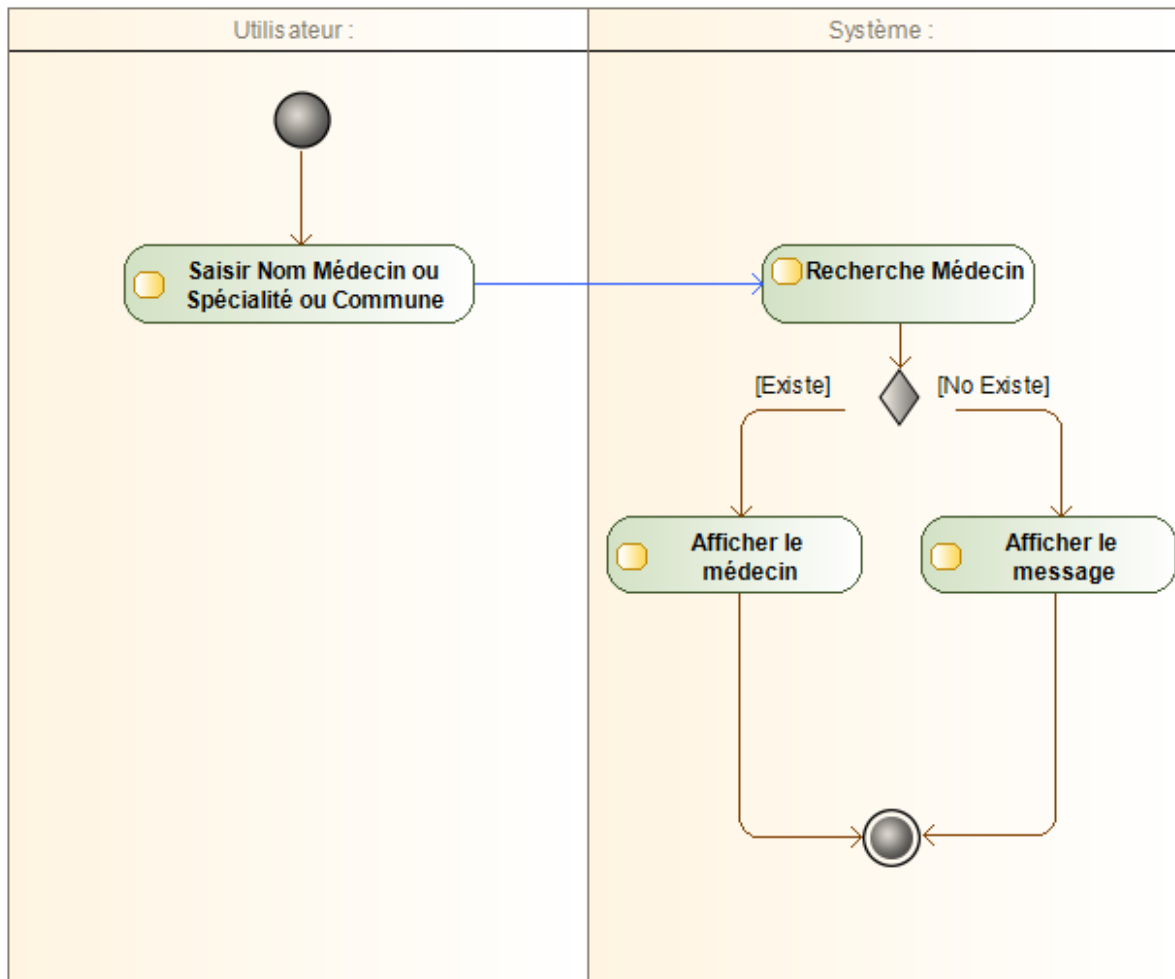


Figure 2.13 : Diagramme d'activité - Recherche médecin -.

Conclusion :

Ce chapitre a présenté d'abord notre problématique suivie de l'environnement de travail qui inclut les outils de développement (logiciels et technologies exploités). Une description des exigences fonctionnelles des différents acteurs de l'application et non fonctionnelles ainsi que la conception UML de l'application sont aussi présentées.

CHAPITRE 3 :
Modélisation et réalisation

Introduction :

Dans ce chapitre, nous présentons l'application réalisée. Pour ce faire les différentes interfaces de l'application sont présentées.

I. Objectif :

Notre projet de fin d'étude de Master 2 a été effectué sous l'encadrement de Mr. MALIKI Fouad. Il consiste à réaliser une application de qualité conçue spécialement pour les téléphones mobiles Android.

Ce projet a pour but de nous initialiser à la plateforme de développement Android et de tirer profit des gadgets que possèdent les Smartphones de nos jours. Le gadget principal dans notre travail est le GPS.

L'application fournit un moyen simple, facile et plus rapide pour aider chaque individu à trouver et localiser un médecin en cas de besoin à partir de son Smartphone. Elle est très utile non seulement afin de subvenir aux besoins des patients, mais elle peut aussi représenter un réel avantage pour les médecins en rapprochant le médecin de ses patients et de le rendre plus disponible et surtout accessible en cas de besoin.

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 et ses perspectives d'évolution sont énormes surtout avec l'essor permanent de la technologie.

II. Description des interfaces de l'application :

1. Logo de l'application :

Nous avons travaillé avec ces deux couleurs qui sont utilisé dans les hôpitaux.



Figure 3.1 : Logo.

2. Capture d'écrans des activités :

Dans ce qui suit nous présentons les différentes interfaces de l'application en citant les détails de chaque imprime écran.



Figure 3.2 : Activité démarrage.

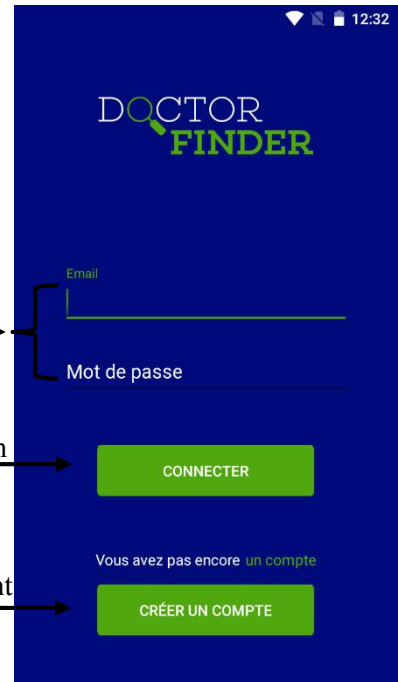


Figure 3.3 : Activité de connexion.

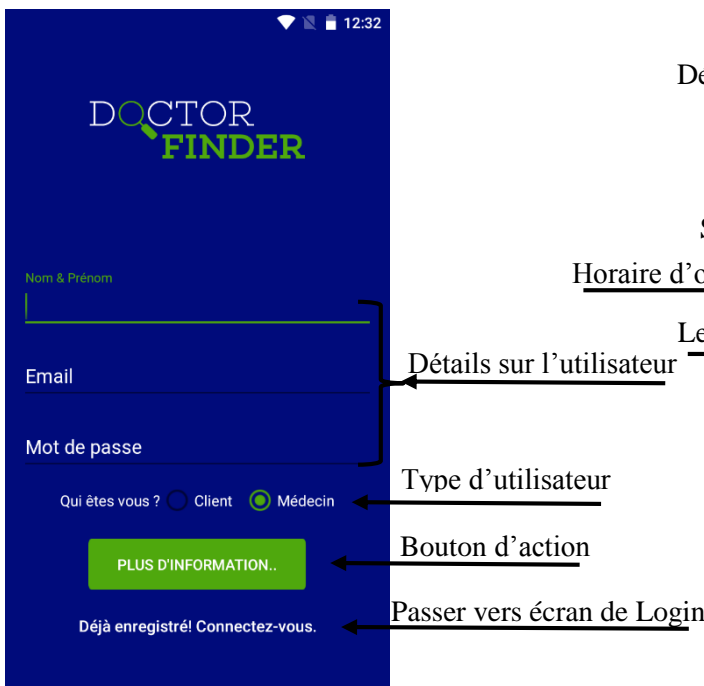


Figure 3.4 : Activité de registration.

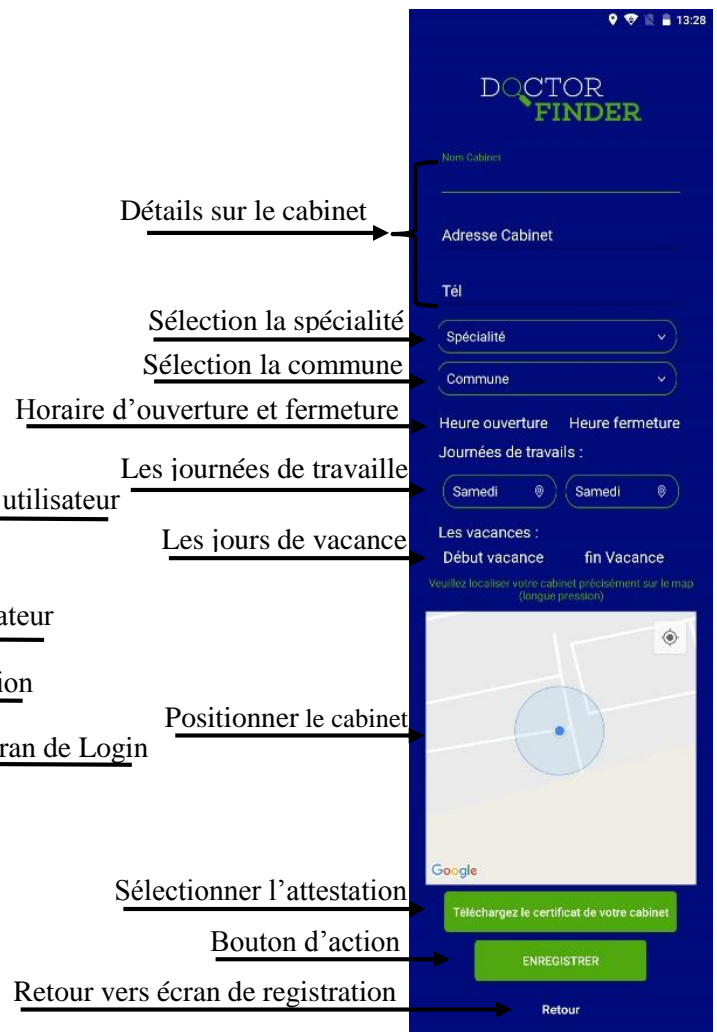
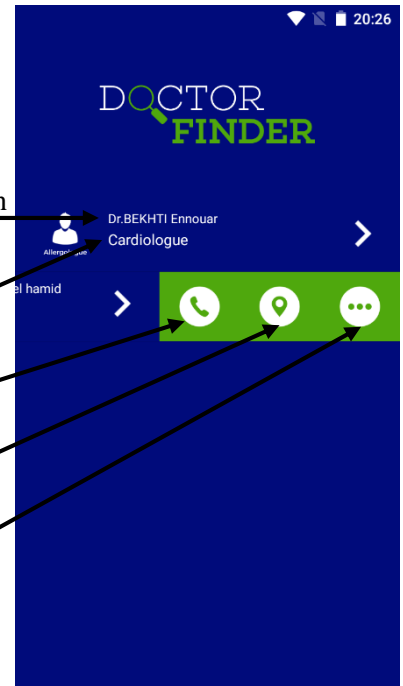


Figure 3.5 : registration plus détails sur le cabinet.



Figure 3.6 : Activité principale.

Les spécialités à choisir



Nom médecin

Spécialité

Accès rapide téléphonique

Accès rapide Map

More information médecin

Figure 3.7 : Liste des médecins d'une spécialité.



Figure 3.8 : fiche d'information.

Information de médecin

Accès téléphonique

Accès téléphonique

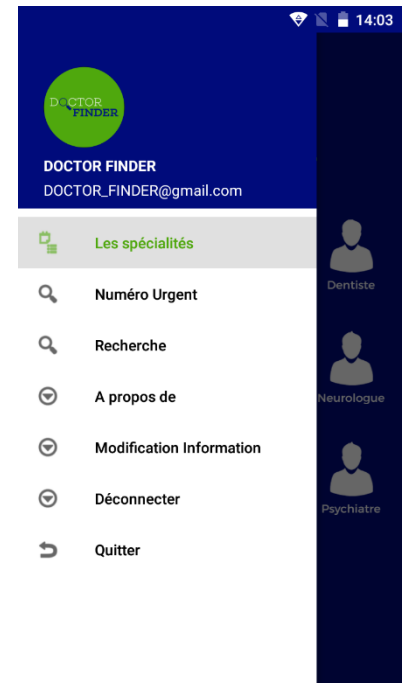


Figure 3.9 : Navigation drawer.

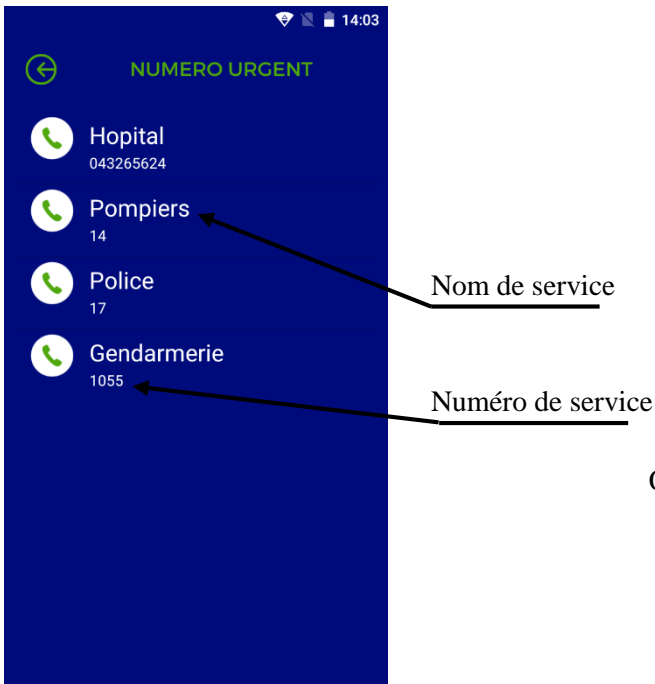


Figure 3.10 : Numéros utile.



Figure 3.11 : Activité recherche.



Figure 3.12 : A propos de.

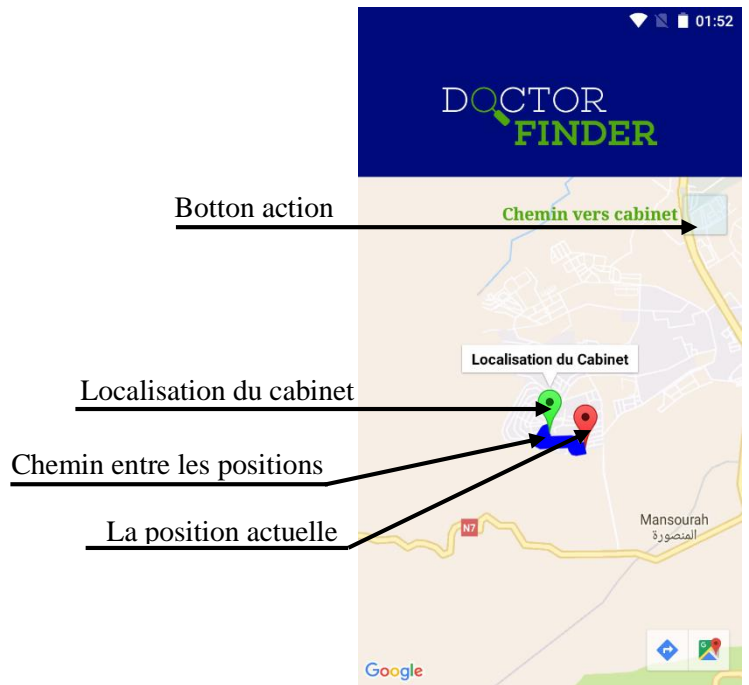


Figure 3.13 : Activité Map.

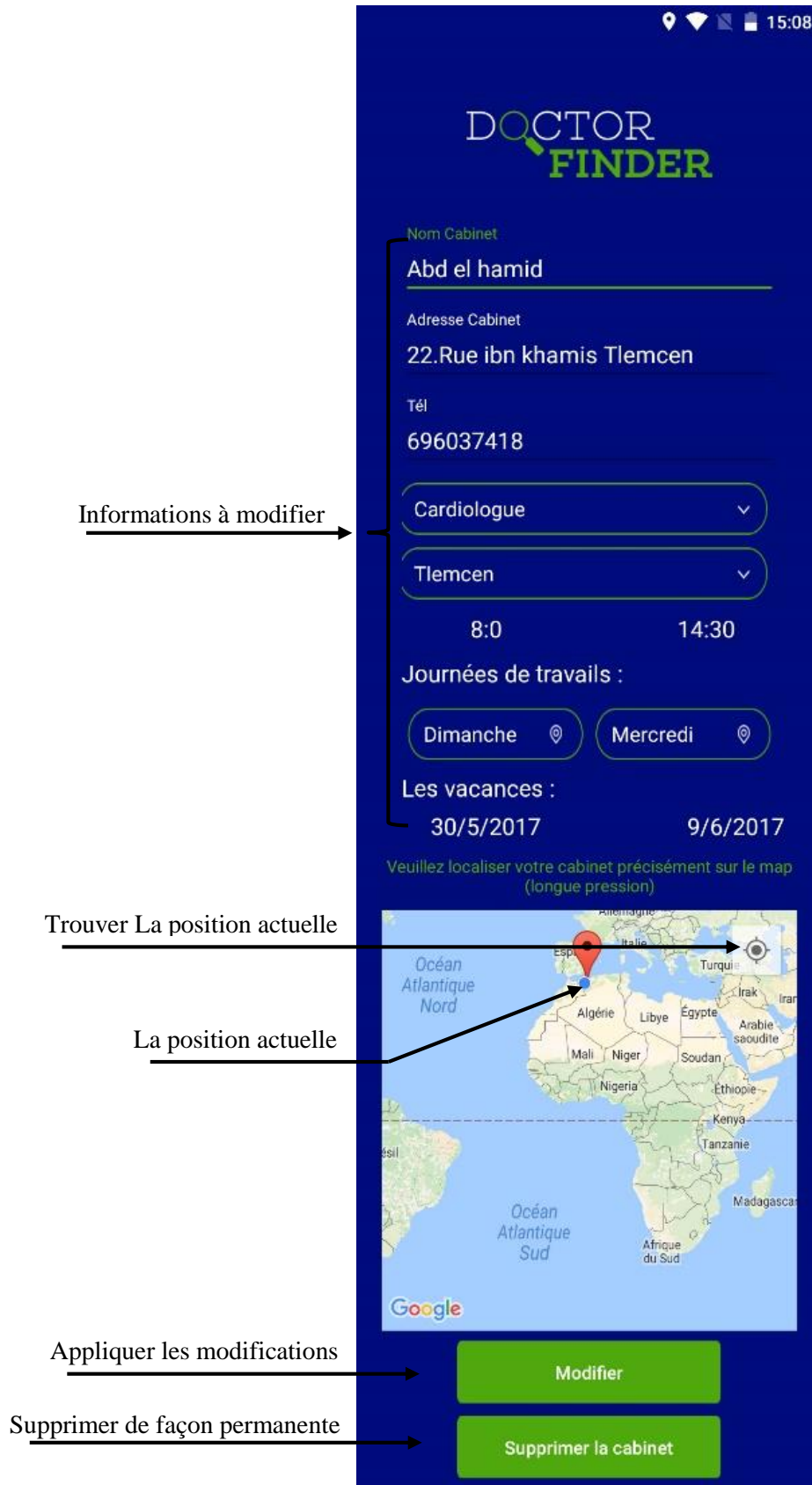


Figure 3.14 : Modifier information.

3. Navigabilité de l'application entre les activités :



Figure 3.15 : Activité démarrage.

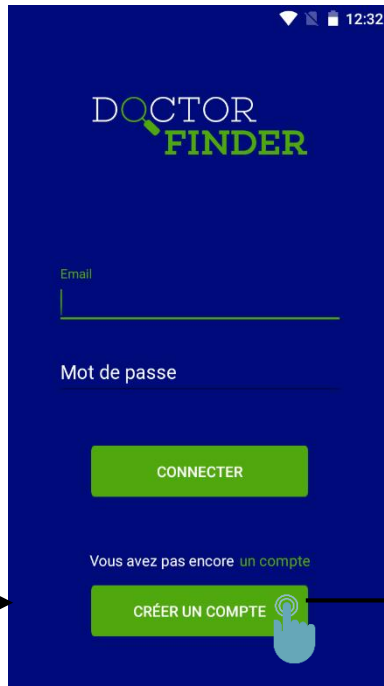


Figure 3.16 : Activité de connexion.

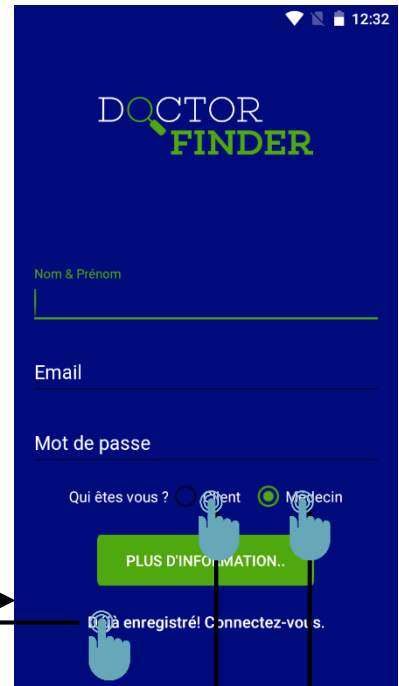


Figure 3.17 : Activité de registration.



Figure 3.18 : Activité principale.



Figure 3.19 : Activité de connexion.

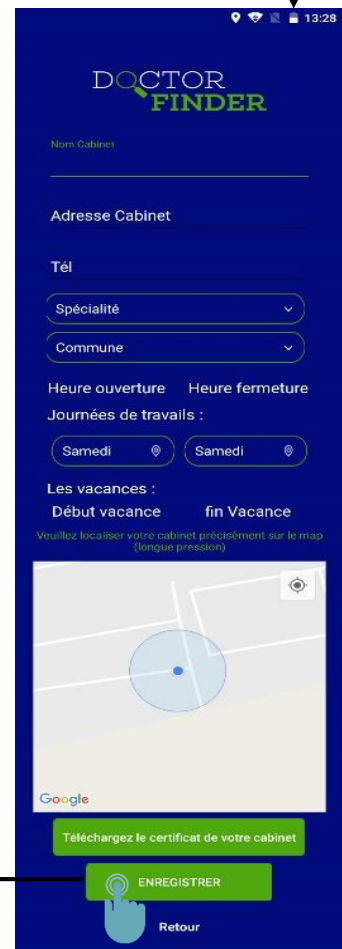


Figure 3.20 : registration plus détails sur le cabinet.



Figure 3.21 : registration plus détails sur le cabinet.

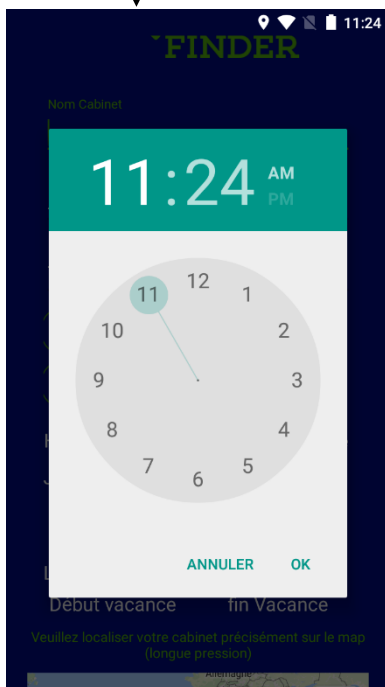


Figure 3.22 : Choix d'horaires.

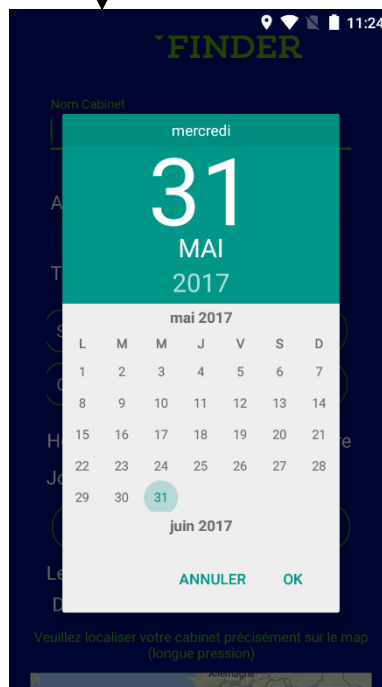


Figure 3.23 : Choix de jour.

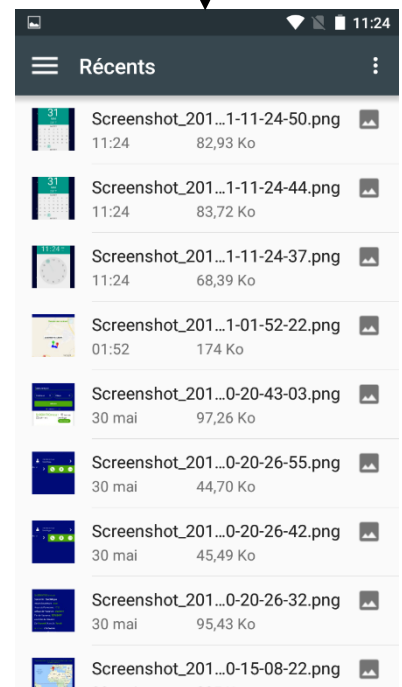


Figure 3.24 : Choix d'attestation.



Figure 3.25 : Activité principale.

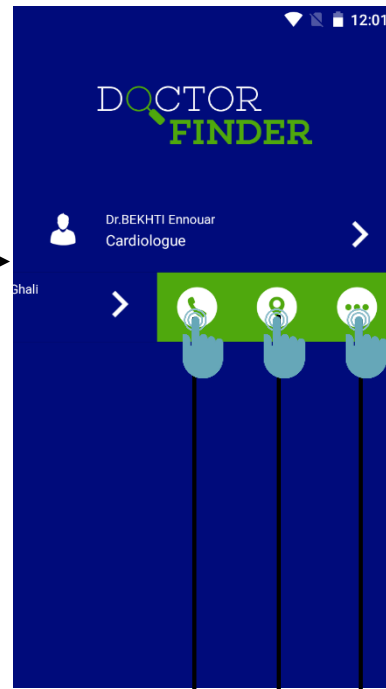


Figure 3.26 : Liste des médecins d'une spécialité.

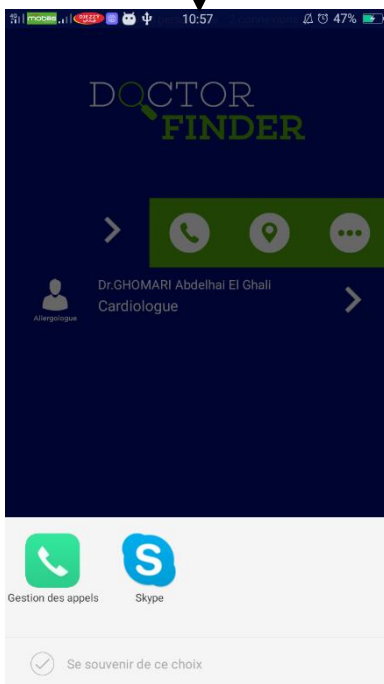


Figure 3.27 : Appeler le médecin.



Figure 3.28 : Activité Map.



Figure 3.29 : fiche d'information.



Figure 3.30 : fiche d'information.

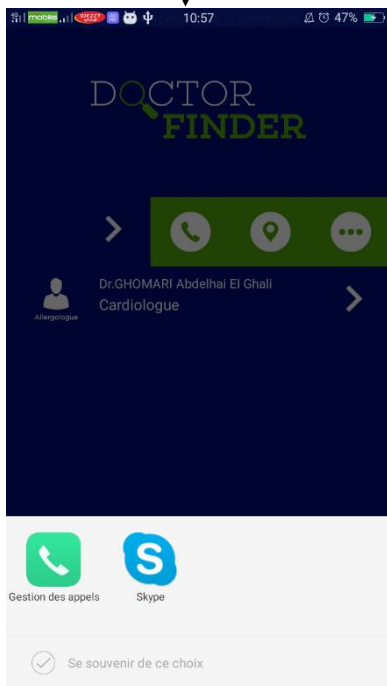


Figure 3.31 : Appeler le médecin.



Figure 3.32 : Activité Map.

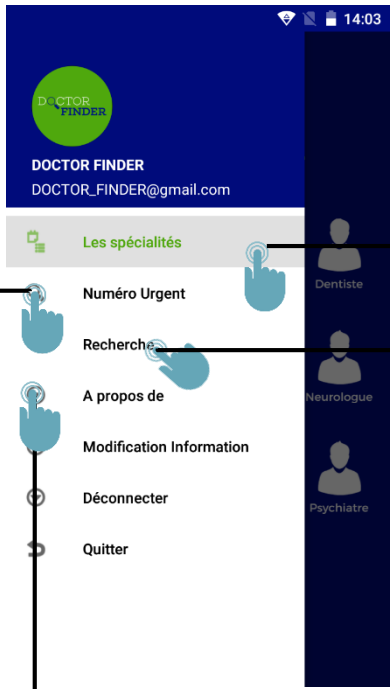


Figure 3.33 : Navigation drawer.



Figure 3.34 : Activité principale.

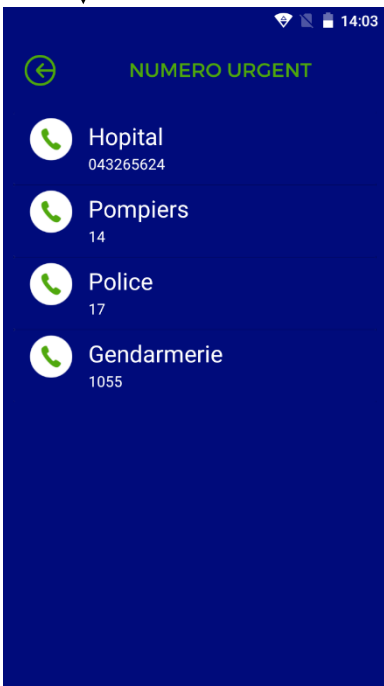


Figure 3.35 : Numéros utile.

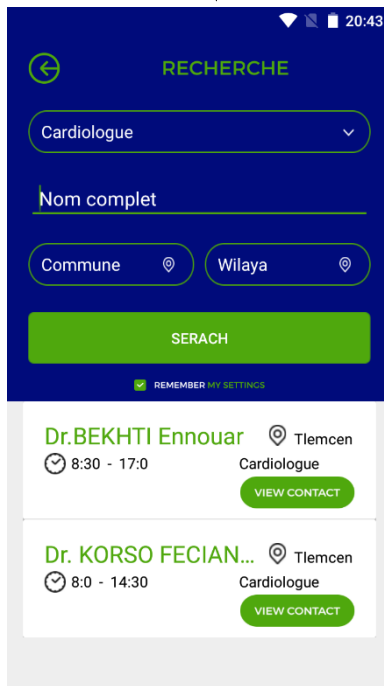


Figure 3.36 : Activité recherche.

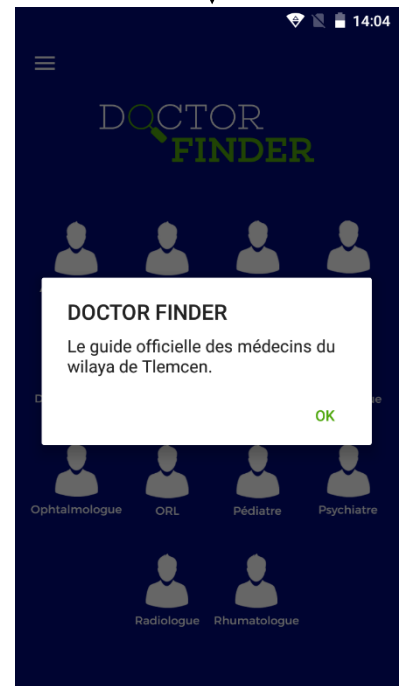


Figure 3.37 : A propos de.

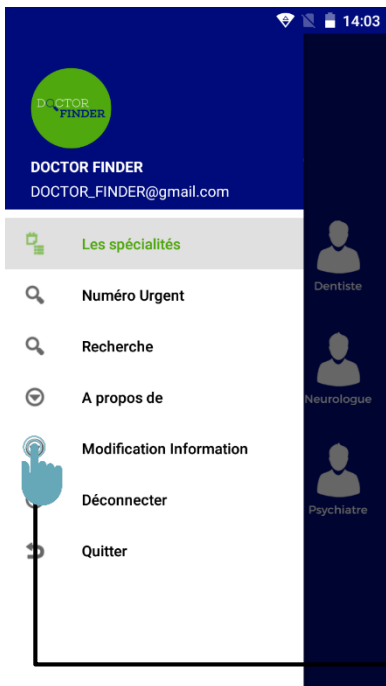


Figure 3.38 : Navigation drawer.



Figure 3.39 : Activité de registration.

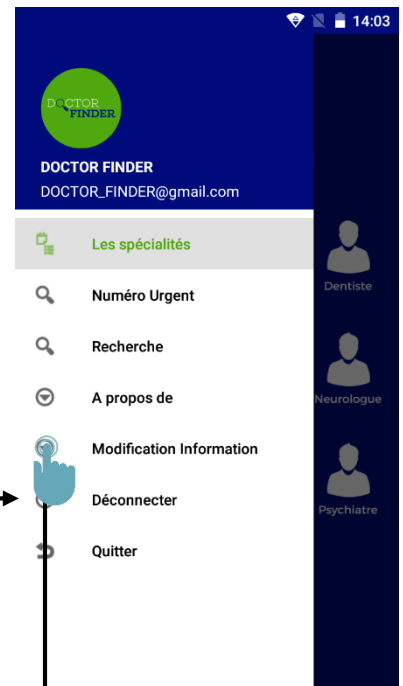


Figure 3.40 : Navigation drawer.



Figure 3.41 : L'affiche de message.



Figure 3.42 : Modifier information.

Conclusion :

Nous avons présenté dans ce chapitre l'objectif de notre projet qui consiste en la réalisation d'une application de localisation des médecins de la wilaya de Tlemcen sous Android. Des captures écrans des activités et de navigation entre elles sont aussi présentées afin de décrire plus en détails l'application réalisée.

Conclusion générale

Ce projet nous a permis de concevoir, développer et intégrer une application mobile sous la plateforme Android qui a fortement occupé le marché des terminaux mobiles.

La réalisation de ce travail est organisée en trois phases, nous avons commencé par une étude préalable de notre projet au cours de laquelle nous avons traité d'une façon générale les différents systèmes d'exploitation et de façons détaillées le système Android en étudiant son architecture, ses versions et ses outils de développements.

La deuxième phase concerne la conception de notre application, nous avons utilisé le langage de modélisation UML pour la partie modélisation et l'environnement de développement Android Studio pour la réalisation.

Enfin, nous avons réussi à réaliser un outil simple et efficace qui aide l'utilisateur à chercher, localiser et contacter les médecins de la wilaya de Tlemcen.

L'outil réalisé est très satisfaisant. Néanmoins, plusieurs perspectives reste envisageable, comme prendre un rendez-vous par mail, gérer les médecins favoris ou ajouter une boîte d'envoie et de réception...

Bibliographie

[En ligne]. Available : <https://www.taktilcommunication.com/blog/applications-mobile/definition-typologie-applications-mobiles.html>

[En ligne]. Available : <https://www.taktilcommunication.com/blog/applications-mobile/definition-typologie-applications-mobiles.html>)

[Mémoire] : Une application Smartphone pour la gestion et la géo-localisation des pharmacies sous Android

[Mémoire] : Conception-et-realisation-dune-application-mobile-sous-Android-de-recherche-de-corps-medicaux

[En ligne]. Available : <http://www.phonandroid.com/toute-l-histoire-et-la-chronologie-d-android-dossier.html>

Livre de Bases de données GEORGES GARDARIN EYROLLES
(www.prism.uvsq.fr/~gardarin/home.html)

[En ligne]. Available : https://jmbruel.github.io/MobileModeling2016-2/plateform_dev.html