

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

*Thème*

**Conception et réalisation d'une  
application mobile sous Android de  
recherche de corps médicaux**

Réalisé par :

- BAGHLI Zineb
- BENMAZOUZ Hakim

*Présenté le 28 Mai 2015 devant la commission d'examen composée de MM.*

- M. I. SMAHI (Encadreur)
- M. MERZOUG (Examineur)
- F. HADJILA (Examineur)

Année universitaire : 2014-2015

# REMERCIEMENT

Nos plus vifs remerciements vont à notre encadreur  
M.SMAHI Mohammed Ismail pour sa compréhension, sa  
disponibilité, son aide et ses précieux conseils qui nous ont  
été très utile pour l'achèvement de ce projet.

Un grand merci à messieurs MERZOUG et HADJILA qui ont  
accepté d'évaluer et d'examiner ce travail.

Un profond merci à toute personne qui a contribué de près  
ou de loin à la réalisation de ce projet.

# DEDICACE

Nous tenons à dédier cet humble travail comme preuve de  
respect, de gratitude et de reconnaissance

A nos chères parents, nos familles et amis

A tous ceux que nous aimons

Pour leurs encouragements, soutiens, patiences et prières.

BAGHLI Zineb

Et

BENMAZOUZ Hakim

**Table des matières**

REMERCIEMENT .....	2
DEDICACE .....	3
Introduction Générale.....	6
Premier chapitre Introduction à Android .....	7
I. Introduction.....	8
II. Que ce que c'est Android ? .....	8
III. Historique .....	8
IV. Version d'Android.....	8
V. L'architecture d'android.....	11
1) Le noyau Linux .....	11
2) Les bibliothèques.....	11
3) Moteur d'exécution Android .....	11
4) Le framework .....	11
5) Applications .....	12
VI. Les outils de développement Android.....	12
• Le SDK Android .....	12
• Le débogueur « ADB » .....	12
• Les IDE « Android Developer Tools » et « Android Studio ».....	12
a) Eclipse .....	12
b) Android Studio .....	12
• La machine virtuelle Dalvik.....	13
• Le runtime Android (ART) .....	13
VII. Conclusion.....	13
Deuxième chapitre Les Bases de données sous Android .....	14
I. Introduction.....	15
II. Composants d'une application Android.....	15
1) Activities (Activités en Français) .....	15
2) Services .....	15
3) Broadcast and Intent Receivers .....	15
4) Content providers .....	15
5) Les Intents : .....	16
III. Cycle de vie d'une activité d'une application Android :.....	16
IV. Les bases de données sous Android .....	17

## Table des matières

1) Base de données .....	17
2) Les bases de données et Android .....	17
a) Mettre en place une BDD externe (MySQL, Oracle...).....	17
b) Mettre en place une BDD interne SQLite .....	18
c) Stocker les données dans un fichier.....	18
3) Comparaison entre une BDD interne et externe.....	18
4) Création d'une application utilisant la base de données SQLite.....	18
a) La base de données .....	19
b) Code java .....	19
c) L'interface graphique .....	20
V. Conclusion.....	21
Troisième chapitre Conception et réalisation de notre Application.....	22
I. Introduction.....	23
II. Conception et analyse de l'application.....	23
1) Objectif.....	23
2) Création de la Base de données.....	23
3) Fonctionnement de notre application .....	24
4) Utilisation de la Google Map : .....	25
5) L'utilisation de l'appel .....	26
6) L'envoi de Mail .....	26
7) AndroidManifest.xml :.....	26
III. Réalisation de l'application.....	27
1) Les outils utilisés.....	27
2) Les limitations .....	27
3) Description de l'application .....	27
a) L'interface principale : .....	28
b) L'interface d'un corps médical :.....	29
c) Les interfaces de Google Map :.....	30
IV. Conclusion :.....	30
Conclusion générale .....	31
Références bibliographiques : .....	32
Liste des figures : .....	33
Liste des tableaux :.....	34
Liste des abréviations .....	35

### Introduction Générale

Les technologies de l'information et de la communication ont été la révolution la plus importante et innovante qui a marqué ces dernières décennies. En effet, loin d'être un phénomène éphémère ces technologies nous ont apporté du confort dans notre vie quotidienne par leurs capacités à traiter l'information dans des délais raisonnables.

Cette révolution a permis l'émergence de la notion de la portabilité et de la mobilité qui permet un accès distant, instantané et un flux sans interruption d'informations. En effet, cela est symbolisé par l'apparition des différents appareils de haute technologie tels que les smartphones et les tablettes qui sont dotés de plusieurs applications pratiques.

Dans ce cadre, et afin de réaliser ce projet de fin d'études, une application proposée permet de pallier les problèmes concernant la question de la recherche des corps médicaux par les patients.

Cet outil propose, avec la version actuelle, solution simple et innovante. Cela a pour vocation de faciliter la recherche des corps médicaux (cabinets médicaux, cliniques, médecins, pharmaciens, radiologues et les laboratoires d'analyses) et de pouvoir les localiser et contacter.

Le présent rapport est organisé comme suit :

Le premier chapitre est consacré à une introduction d'android, citant son historique, ses versions, son architecture et ses outils de développements.

Quant au deuxième chapitre, il est dédié aux bases de données sous Android, où il y aura en premier lieu les composants d'une application android, et le cycle de vie d'une activité d'une application ensuite les bases de données sous Android, et à la fin de ce chapitre il y aura un guide pour la création d'une application basique utilisant une base de données SQLite.

Et le dernier chapitre, sera destiné à la conception et la réalisation de notre application pour ce projet de fin d'études.

**Premier chapitre**  
**Introduction à Android**

## I. Introduction

Dans le monde actuel, les smartphones et tablettes font partis de notre vie quotidienne, en la rendant plus facile et commode, ils sont équipés d'applications qui sont conçues pour des plateformes mobiles et utilisés pour des services d'information, de medias sociaux, de jeux etc...

Un smartphone est doté d'un système d'exploitation et cela varie de sa marque, celui qui est le plus répandu sur le marché est « Android » (grâce à son open source). Dans ce chapitre nous nous intéressons au développement mobile sous Android, en détaillant son historique, ces caractéristiques ainsi que son architecture, ses avantages et inconvénients et enfin la concurrence sur le marché.

## II. Que ce que c'est Android ?

Android est une plateforme complète pour appareil mobile (téléphone, tablette, TV, auto...). Elle est composée d'un système d'exploitation, de librairies « middleware », et d'un ensemble d'applications : un navigateur, un calendrier, un client mail ...

## III. Historique

Android a été conçu par une startup éponyme spécialisée dans le développement d'applications mobiles en 2003 puis a été racheté par google en 2005 et annoncé officiellement le 15 novembre 2007.

Afin de promouvoir Android, google a su fédérer autour de lui une trentaine de partenaires réunis au sein de l'OHA<sup>1</sup>.

## IV. Version d'Android

Android a connu plusieurs versions, et chaque nouvelle version sert à corriger des bugs et ajouter de nouvelles fonctionnalités, ayant chacune d'elle droit à un nom de code sur la thématique des desserts suivant une logique alphabétique.

---

<sup>1</sup> OHA : est une alliance internationale constituée de compagnies ne faisant pas partie du même secteur.



Dans le tableau suivant, il y a le nom de code de chaque version ainsi que sa date de sortie avec les caractéristiques majeures ajoutées à chaque une d'elles [1] :

Nom de code	Version	Date de sortie	Majeurs caractéristiques
<b>Apple pie</b>	1.0	11 Novembre 2007	<ul style="list-style-type: none"> <li>• Téléchargement et mises à jour des applications via Android Market.</li> <li>• Support du Wi-Fi et du Bluetooth.</li> </ul>
<b>Bananas split</b>	1.1	22 Octobre 2008	<ul style="list-style-type: none"> <li>• "Afficher" &amp; "Cacher" le pavé numérique, inclus dans le menu d'appel</li> <li>• Support pour sauvegarder les fichiers attachés aux MMS</li> </ul>
<b>Cupcake</b>	1.5	30 Avril 2009	<ul style="list-style-type: none"> <li>• Support pour les Widgets, qui permettent d'accéder rapidement à certaines informations de l'application à laquelle ils sont rattachés.</li> <li>• Ajout d'une option de rotation automatique</li> </ul>
<b>Donut</b>	1.6	15 Septembre 2009	<ul style="list-style-type: none"> <li>• Possibilité pour les développeurs d'intégrer leurs contenus dans les résultats de recherche</li> <li>• Amélioration de la rapidité dans la recherche et les applications utilisant la caméra</li> </ul>
<b>Eclair</b>	2.0	26 Octobre 2009	<ul style="list-style-type: none"> <li>• Performance matériel optimisée</li> <li>• Support de plus de taille d'écran et résolutions</li> </ul>
<b>Froyo</b>	2.2	20 Mai 2010	<ul style="list-style-type: none"> <li>• Amélioration de la vitesse d'exécution</li> <li>• Partage de connexion USB</li> </ul>
<b>Gingerbread</b>	2.3	06 Décembre 2010	<ul style="list-style-type: none"> <li>• Support du NFC et de nouveaux capteurs (comme le gyroscope et le baromètre)</li> <li>• Amélioration de la gestion de l'alimentation et du contrôle des applications</li> </ul>

<b>Honeycomb</b>	3.0	22 Février 2011	<ul style="list-style-type: none"> <li>• Meilleur support des tablettes</li> <li>• Bureau tridimensionnel avec widgets améliorés</li> </ul>
<b>Ice Cream Sandwich</b>	4.0	19 Octobre 2011	<ul style="list-style-type: none"> <li>• Amélioration du texte d'entrée et du correcteur orthographique</li> <li>• Ajout du WI-FI direct</li> </ul>
<b>Jelly Bean</b>	4.1	09 Juillet 2012	<ul style="list-style-type: none"> <li>• Amélioration de la vitesse d'exécution et de la gestion d'appareil photo</li> <li>• Android Beam (Transfert de données rapide : NFC + Bluetooth)</li> </ul>
	4.2	13 Novembre 2012	<ul style="list-style-type: none"> <li>• Widgets en mode verrouillé</li> <li>• Multi-utilisateurs pour les tablettes</li> </ul>
	4.3	24 Juillet 2013	<ul style="list-style-type: none"> <li>• Ajout du support pour la résolution 4K</li> <li>• Support de Bluetooth basse consommation (BLE) et du profil Bluetooth (AVRCP) 1.3</li> </ul>
<b>KitKat</b>	4.4	31 Octobre 2013	<ul style="list-style-type: none"> <li>• Nouvelle interface translucide</li> <li>• Amélioration du système de notification et des performances</li> </ul>
<b>Lollipop</b>	5.0	03 Novembre 2014	<ul style="list-style-type: none"> <li>• Nouvelle interface de design « Material design »</li> <li>• Amélioration de la rapidité et de la gestion de la batterie</li> </ul>
	5.1	09 Mars 2015	<ul style="list-style-type: none"> <li>• Protection par blocage en cas de perte ou vol</li> <li>• Amélioration de la stabilité et des performances</li> </ul>

Tableau 1 : Les différentes versions d'Android

## V. L'architecture d'android

La figure suivante schématise l'architecture d'android [2] :



**Figure 1 : Les principaux composants du système d'exploitation android**

Ainsi, les principaux composants de ce système sont :

### 1) 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.

### 2) Les bibliothèques

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

### 3) Moteur d'exécution Android

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

### 4) Le framework

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

### 5) 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.

## VI. Les outils de développement Android

Les différents outils de développement Android sont :

### •Le SDK Android

Le Kit de développement logiciel Android (Android SDK) contient les outils nécessaires pour créer, compiler et déployer les applications Android. La plupart de ces outils sont en ligne de commande.

### •Le débogueur « ADB »

Le SDK Android contient un débogueur appelé « Android debug bridge » ou aussi « adb », qui permet de connecter un appareil Android virtuel ou réel, dans le but de gérer le périphérique ou de déboguer votre application.

### •Les IDE « Android Developer Tools » et « Android Studio »

Google propose deux environnements de développement intégrés (IDE) pour développer de nouvelles applications.

#### a) Eclipse

Les outils de développement Android sont basés sur l'IDE Eclipse. ADT est un ensemble de composants (plug-ins), qui étendent l'IDE Eclipse avec des capacités de développement Android.

#### b) Android Studio

Google propose également cet IDE appelé pour la création d'applications Android quel que soit le terminal sous-jacent (smartphone, tablette, montre, TV...) et qui est basé sur l'IDE IntelliJ.

Le tableau qui suit, compare entre les deux IDE :

	<b>Eclipse</b>	<b>Android Studio</b>
Facilité d'installation	Moyen	Simple
Langue	Nombreuses	Anglais
Performance	Peut être lourd	Rapide
Système de built	Ant	Gradle
Génération de variante et de multiple APK	Non	Oui
Android Code completion et refactoring	Base	Avancé
Editeur d'interface graphique	Oui	Oui
Signature d'APK et gestion de Keystore	Oui	Oui
Support NDK	Oui	A venir

**Tableau 2** Tableau comparatif entre Eclipse et Android Studio

### •La machine virtuelle Dalvik

Dalvik est une machine virtuelle incorporée dans le système d'exploitation Android. Destinée à permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité (peu d'espace mémoire et peu de puissance de calcul).

### •Le runtime Android (ART)

Les dernières versions d'Android introduisent une nouvelle machine, le runtime Android. le moteur qui permet l'exécution des applications pour Android.

## VII. Conclusion

Dans ce chapitre, nous avons étudié la plateforme Android en voyant son historique, ses versions, son architecture et ses outils de développements.

**Deuxième chapitre**  
**Les Bases de données sous**  
**Android**

## I. Introduction

Pour la plupart des applications Android, le stockage de données est quelques choses d'inévitable, pour cela le système d'exploitation Android fournit différentes méthodes de sauvegarde d'informations, y parmi l'utilisation de la base de données SQLite qui permet d'enregistrer les données de façon rapide et flexible.

Dans ce chapitre nous parlons d'abord des composants d'une application android et le cycle de vie d'une activité d'une application android, ensuite des bases de données sous android. A la fin, la création d'une application utilisant la base de données SQLite.

## II. Composants d'une application Android

Une application Android est composée des éléments basiques suivants [3] :

### 1) Activities (Activités en Français)

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

### 2) Services

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

### 3) Broadcast and Intent Receivers

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

### 4) Content providers

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

### 5) Les Intents :

Les composantes Android communiquent via des messages système que l'on appelle intent. Ils sont émis par le terminal pour prévenir les différentes applications du déclenchement d'évènements.

### III. Cycle de vie d'une activité d'une application Android :

Pour développer une application android, il faut comprendre le cycle de vie d'une activité qui est illustré par la figure si dessous [4] :

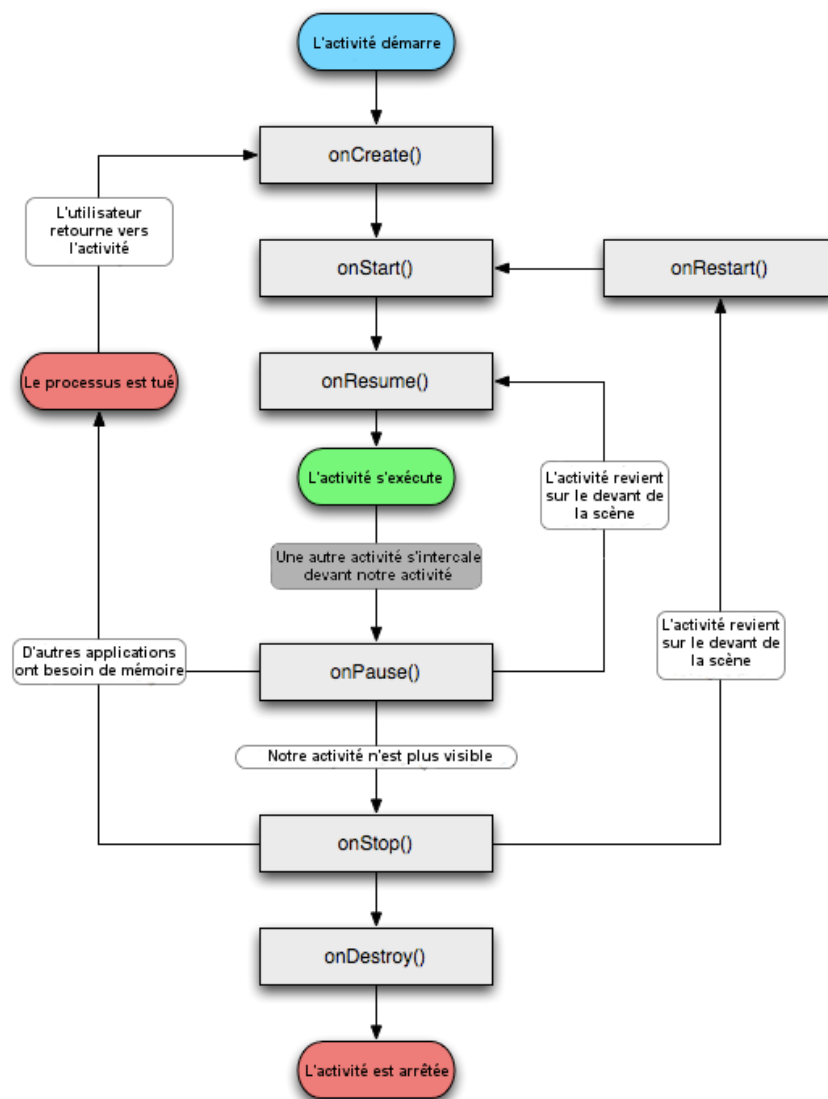


Figure 2: Cycle de vie d'une activité



Les différentes méthodes qu'on peut trouver dans le cycle de vie d'une activité sont les suivantes :

- **onCreate () /onDestroy ()**: permet de gérer les opérations à faire avant l'affichage de l'activité, et lorsqu'on détruit complètement l'activité de la mémoire. On met en général peu de code dans onCreate () afin d'afficher l'activité le plus rapidement possible.
- **onStart () / onStop ()**: ces méthodes sont appelées quand l'activité devient visible/invisible pour l'utilisateur.
- **onPause () / onResume ()**: une activité peut rester visible mais être mise en pause par le fait qu'une autre activité est en train de démarrer, par exemple B. onPause() ne doit pas être trop long, car B ne sera pas créé tant que onPause() n'a pas fini son exécution.
- **onRestart ()**: cette méthode supplémentaire est appelée quand on relance une activité qui est passée par onStop (). Puis onStart () est aussi appelée. Cela permet de différencier le premier lancement d'un relancement.

## IV. Les bases de données sous Android

### 1) Base de données

Une base de données est un outil permettant de stocker et de retrouver l'intégralité de données brutes ou d'informations, celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles. Dans la très grande majorité des cas, ces informations sont très structurées.

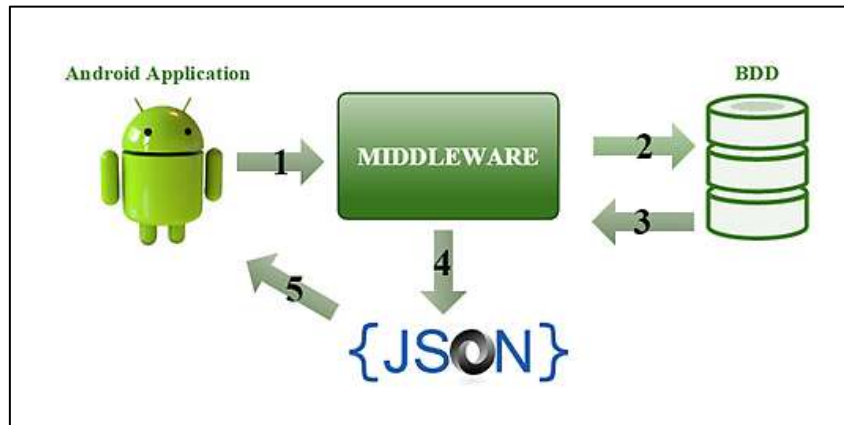
### 2) Les bases de données et Android

Concernant le stockage des Données sous Android, il existe plusieurs possibilités [5]:

#### a) Mettre en place une BDD externe (MySQL, Oracle...)

Sous android, la méthode la plus répandue pour établir une connexion entre une bdd externe et une application, est de passer par un middleware qui va organiser, adapter et traiter les échanges entre l'application et la BDD. Ces échanges sont possibles grâce à des

Web Services, avec des langages d'échanges adaptés tels que JSON, ou XML.



**Figure 3: Connexion avec une BDD externe**

**b) Mettre en place une BDD interne SQLite**

SQLite une BDD SQL open source qui stocke les données dans un fichier texte sur un dispositif. Elle est intégrée dans tous les appareils Android cela dit une base de données SQLite ne nécessite aucune configuration ou mise en place. Une fois la structure de la bdd est définie, elle sera automatiquement gérée par Android.

**c) Stocker les données dans un fichier**

C'est la création d'un fichier de données à la compilation. Cependant les données contenues par ce fichier seront accessibles uniquement en lecture.

**3) Comparaison entre une BDD interne et externe**

Dans le tableau suivant apparaissent les différences entre la bdd interne et externe :

Base de données externe	Base de données interne
L'espace de stockage illimité	L'espace de stockage limité
L'impossibilité d'accéder à la base de données si le smartphone n'est pas connecté	La possibilité d'accéder à la base de données si le smartphone n'est pas connecté à Internet
Le temps de réponse supérieur	Le temps de réponse inférieur

**Tableau 3: Comparaison entre BDD interne et externe**

**4) Création d'une application utilisant la base de données SQLite**

Pour cet exemple, nous allons créer une application « SqliteApp » qui donne le droit à l'utilisateur d'insérer des données et qui vont être stockés dans une bdd interne SQLite.

**a) La base de données**

Nous aurons une base de données appelée « Informations.db », qui comportera une seule table « info\_ table». Ses champs sont (ID, Nom, Prenom, Email, Telephone).



Figure 4: La table "Info\_table" de notre BDD

**b) Code java**

Nous allons créer deux classes :

**La première classe «DatabaseHelper» :** qui est une classe d’assistance pour gérer la création de bases de données, la gestion des versions et où on trouvera la méthode ajout qui va ajouter les données entrées par l’utilisateur dans la bdd.

```

public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, 1);
}
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table " + TABLE_NAME + " (ID INTEGER PRIMARY KEY AUTOINCREMENT,NOM " +
        "TEXT,PRENOM TEXT,EMAIL TEXT,TELEPHONE INTEGER)");
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
    onCreate(db);
}
public boolean ajout(String nom,String prenom,String email,String telephone) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_2,nom);
    contentValues.put(COL_3,prenom);
    contentValues.put(COL_4,email);
    contentValues.put(COL_5,telephone);
    long result = db.insert(TABLE_NAME,null ,contentValues);
    if(result == -1)
        return false;
    else
        return true;
}
    
```

Figure 5: Les méthodes de la classe "DatabaseHelper".

**La deuxième classe la « MainActivity »:** qui est la classe principale et dans laquelle on implémentera la méthode AjouterInfo, qui va insérer les données dans la bdd si l’utilisateur cliquera sur le bouton « ajoute » et affichera le résultat dans des Toasts.

```

public void AjouterInfo() {
    btnAjouter.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                boolean isInserted = myDb.ajout(editNom.getText().toString(),
                    editPrenom.getText().toString(),
                    editEmail.getText().toString(),
                    editTelephone.getText().toString());
                if (isInserted == true)
                    Toast.makeText(MainActivity.this,
                        "Informations Ajoutées", Toast.LENGTH_LONG).show();
                else
                    Toast.makeText(MainActivity.this,
                        "Informations Non Ajoutées", Toast.LENGTH_LONG).show();
            }
        }
    );
}
    
```

Figure 6: Code source de la méthode "AjouterInfo".

c) L'interface graphique

Nous allons mettre 4 « editText » pour insérer: le nom, le prénom, l'email et le téléphone, et un bouton « ajouter ».

Notre interface graphique sera :



Figure 7: l'interface graphique de l'application.

Et finalement, nous avons créés une application qui ajoutera des informations utilisant une BDD SQLite.

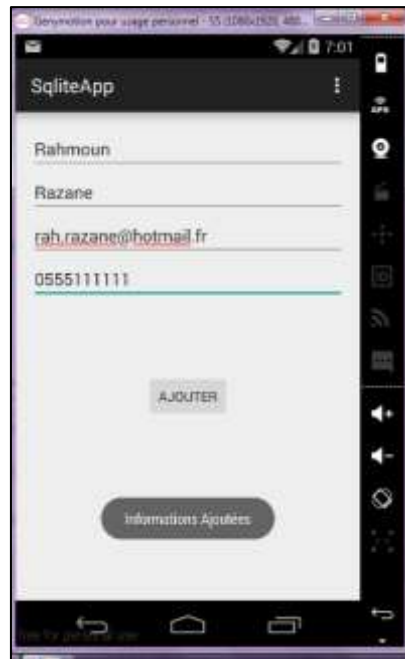


Figure 8: l'application "SqliteApp".

## V. Conclusion

Dans ce chapitre, nous avons parlé d'abord des composants d'une application android et le cycle de vie d'une activité d'une application android, ensuite des bases de données sous android en citant les différentes façon de relier une base de données et Android et en comparant entre un bdd externe et une bdd interne et nous avons terminé ce chapitre avec la création d'une application utilisant la base de données SQLite.

**Troisième chapitre**  
**Conception et réalisation de**  
**notre Application**

## I. Introduction

Tous les jours, les gens sont amenés à chercher à joindre une ou plusieurs structures des corps médicaux. Vu la grandeur de la ville, et les différents emplacements des dites structures, il leur est difficile de mémoire de s'auto-orienter vers le service adéquat.

Pour cela, nous avons pensé à développer pour notre projet de fin d'étude, une application android qui permettra aux utilisateurs de chercher les corps médicaux, de voir leurs emplacements GPS et de pouvoir les contacter.

Dans ce chapitre nous allons voir la conception de notre application en commençant par notre objectif ensuite la création de la base de données, le fonctionnement de l'application et l'utilisation de Google Map, l'appel et l'envoi d'email et en finissant avec la réalisation de l'application en voyant les outils de développement utilisés, les limitations et les interfaces graphiques de l'application.

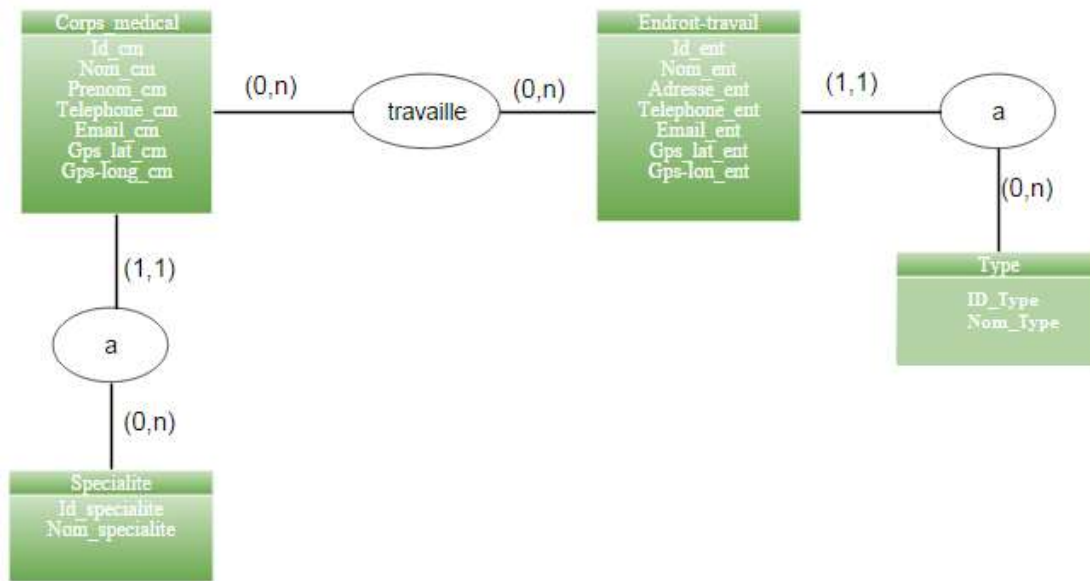
## II. Conception et analyse de l'application

### 1) Objectif

Notre objectif est de créer une application Android qui aidera les utilisateurs à rechercher les corps médicaux situés à Tlemcen (qu'ils soient cabinets médicaux, cliniques, médecins, pharmaciens, radiologues de même que laboratoires d'analyses) en utilisant une Google Map pour pouvoir voir la localisation de ces corps médicaux, ainsi d'avoir la possibilité de les contacter par mail ou par appel téléphonique.

### 2) Création de la Base de données

Pour notre application, nous avons choisi d'utiliser une base de données interne SQLite qu'on appellera « Recherche.db » et qui comportera 4 tables, le MCD suivant nous décrit sa sémantique :



**Figure 9: le MCD de la BDD**

Et ainsi nous obtenons après avoir fait les modifications nécessaires le MLD suivant:

- Corps\_médical (id\_cm, nom\_cm, prenom\_cm, telephone\_cm, email\_cm, Gps\_lat\_cm, Gps\_lon\_cm, #~id\_specialite)
- Spécialité (id\_specialite, nom\_specialite)
- Endroit\_travail (id\_ent, nom\_ent, prenom\_ent, telephone\_ent, email\_ent, Gps\_lat\_ent, Gps\_lon\_ent, #ID\_type)
- Type (ID\_type, Nom\_type)
- Travaille (#id\_cm , #id\_ent)

Dès lors, la création de la Base de données « Recherche.db » est possible.

### 3) Fonctionnement de notre application

Notre application est reliée à une base de données interne.

Dans la première interface de l'application, il y aura seulement des boutons qui nous mèneront vers les interfaces secondaires.

Dans chaque une des interfaces secondaires, on fera appel à cette base de données, on affichera les informations personnelles des corps médicaux. De même, on aura la possibilité de passer à l'interface de Google Map.

Dans l'interface de Google Map, on mettra des markers qui vont localiser les corps médicaux sur la Map.



La figure suivante, résume via un diagramme de cas d'utilisation, l'essentiel de notre travail :

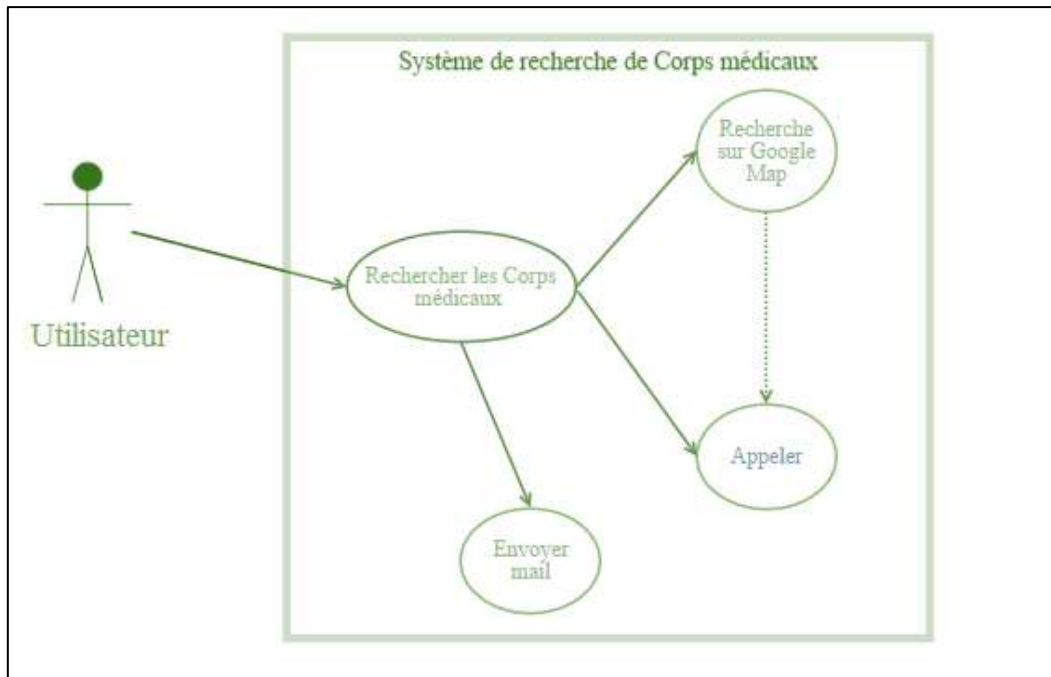


Figure 10: Diagramme de cas d'utilisation générale.

#### 4) Utilisation de la Google Map :

Pour intégrer une Google Map dans une application android, nous avons besoin d'une clé de l'API qu'on obtient de google.



Figure 11: La clé de Google Map

De plus, nous devons ajouter des permissions dans le fichier « AndroidManifest.xml ».

Pour localiser un endroit dans une Google Map, il existe les markers<sup>2</sup> qui désignent dans notre application, l'emplacement des différents corps médicaux, et qui donnent la possibilité en cliquant sur eux d'appeler.

<sup>2</sup>Marker : Google Map Maker est une manière pour Google de modifier ses cartes depuis juin 2008.

Le code suivant est celui de la fonction qui permet d'afficher la petite fenêtre du marker et d'appeler le corps médical :

```
@Override
public void onInfoWindowClick(Marker marker) {
    Intent intent = new Intent(this, information.class);
    intent.putExtra("snippet", marker.getSnippet());
    intent.putExtra("title", marker.getTitle());
    intent.putExtra("position", marker.getPosition());

    Intent call = new Intent(Intent.ACTION_DIAL);
    call.setData(Uri.parse(marker.getSnippet()));
    startActivity(call);
    //startActivity(intent);
}
```

Figure 12: Code Source de la fenêtre du marker.

### 5) L'utilisation de l'appel

Pour pouvoir appeler un corps médical, faut ajouter des permissions dans le fichier « AndroidManifest.xml ».

### 6) L'envoi de Mail

Pour envoyer un mail, google ne fournit pas une API directe, donc nous devons appeler le client de messagerie existant de l'appareil.

### 7) AndroidManifest.xml :

Dans la figure suivante, nous avons le « AndroidManifest.xml » de notre application avec toutes les permissions ajoutés :

```

<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=" AIzaSyC1aH2OXc0iAlpho_3fkiSicK-z2GrXi94" />

<activity
    android:name=".MainActivity"
    android:label="ppfe" >
    <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 13: Les permissions ajoutées dans l'AndroidManifest.xml"

### III. Réalisation de l'application

#### 1) Les outils utilisés

Pour développer cette application, nous avons utilisés :

- L'environnement de développement Android Studio.
- Le SDK de l'Android.
- ADT
- Gliffy pour la conception des diagrammes et des figures.
- Les terminaux mobiles : LG G2 et Sony T3.

#### 2) Les limitations

- Notre base de données ne contient que 10 données pour chaque corps médical (en tout, nous avons 60 corps médicaux ajoutés à l'application).
- Notre application est compatible seulement avec les smartphones qui peuvent supporter les versions entre API18 et API22.

#### 3) Description de l'application

Nous allons présenter les principales interfaces graphiques de notre application ;

**a) L'interface principale :**

Dans l'interface d'accueil, il s'affichera les boutons de tous les corps médicaux (cabinets médicaux, cliniques, médecins, laboratoires d'analyses, pharmaciens et radiologues), on a la possibilité d'accéder à n'importe lequel des boutons que l'on souhaite.



**Figure 14: l'interface d'Accueil de l'application.**

**b) L'interface d'un corps médical :**

Comme exemple, nous allons choisir le bouton « MEDECINS » l'application nous orientera vers une nouvelle interface graphique comme la figure suivante :



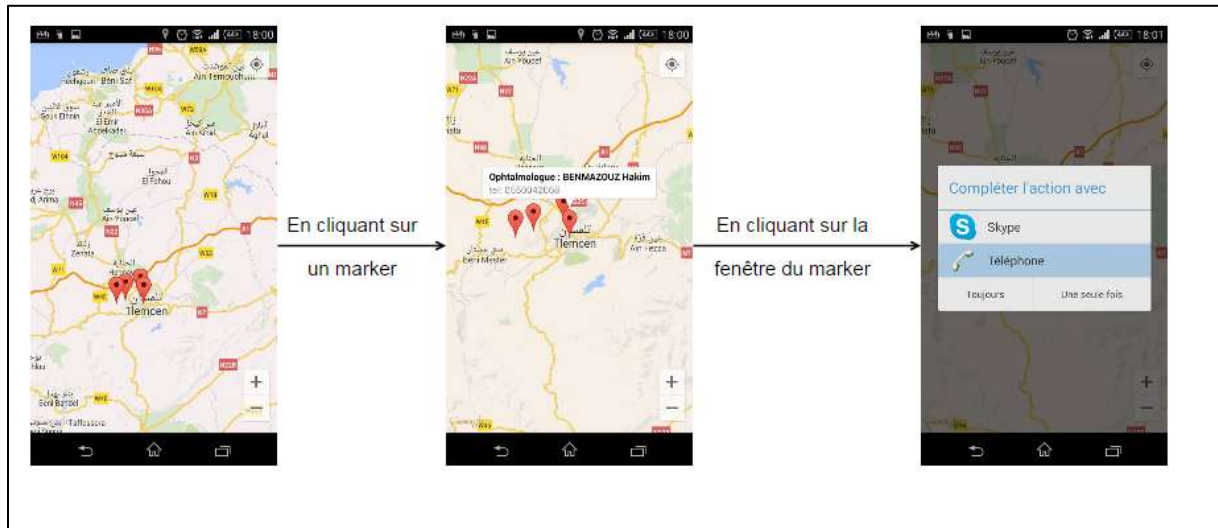
**Figure 15: Interface graphique des "MEDECINS".**

Et qui aura :

- **un bouton « Voir la MAP »** qui en cliquant nous mènera vers une nouvelle interface graphique de Google Map.
- **un « text View »** qui indique que c'est la liste des médecins de Tlemcen.
- **un « editText »** relié à la bdd qui sert à rechercher les médecins par spécialité.
- **une « listview »** qui affiche les informations personnelles de chaque médecin à partir de la bdd (son nom complet, son adresse, sa spécialité et l'endroit où il travaille) de plus il y a deux boutons le premier pour appeler ce médecin et le deuxième pour lui envoyer un mail.

**c) Les interfaces de Google Map :**

La Google Map contiendra des markers qui représentent les médecins de Tlemcen, en zoomant, et en cliquant sur le marker le nom du médecin, sa spécialité ainsi que son numéro de téléphone s'afficheront sur une petite fenêtre et en cliquant sur elle, l'appel s'effectuera automatiquement.



**Figure 16: les Interfaces de Google Map.**

**IV. Conclusion :**

Dans ce chapitre, nous avons parlé dans un premier lieu de l'objectif de notre application, de la création de la base de données, le fonctionnement de l'application et l'utilisation de Google Map, l'appel et l'envoi d'email.

Et dans un second lieu nous avons parlé de la réalisation de l'application en voyant les outils de développement utilisés, les limitations et les interfaces graphiques de l'application.

### Conclusion générale

Ce projet nous a permis d'avoir une idée plus claire et plus précise sur le développement sous Android, ce système d'exploitation qui a envahi le marché des terminaux mobiles.

Tout d'abord, nous avons traité d'une façon détaillée ce système en voyant son historique, ses versions, son architecture et ses outils de développements.

Puis, nous avons étudié l'utilisation des bases de données sous Android, les différentes façons de connecter une application à une BDD, et nous avons terminés ce chapitre avec la création d'une application utilisant les bases de données interne.

Et notre dernier chapitre concerne la conception et la mise en œuvre de notre application qui aide l'utilisateur à chercher et à contacter les corps médicaux à Tlemcen qui a été développée avec Android Studio.

L'objectif principal a été atteint, mais tout de même l'amélioration de notre application reste envisageable, comme l'ajout d'un moteur de recherche, l'utilisation d'une base de données externe ou l'ajout des administrateurs qui auront droit d'ajouter les médecins manuellement.

**Références bibliographiques :**

[1] <http://socialcompare.com/fr/comparison/android-versions-comparison>

[2] <http://openclassrooms.com/courses/creez-des-applications-pour-android/l-architecture-d-android>

[3] :N.BENBOURAHLA,"Android 4 les fondamentaux fe developpement d'application java",2012 .

[4] <http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/cours-android.pdf>

[5] <http://tutorielandroid.francoiscolin.fr/bdd.php>



## Liste des figures :

<b>Figure 1:</b> Les principaux composants du système d'exploitation android .....	11
<b>Figure 2:</b> Cycle de vie d'une activité .....	16
<b>Figure 3:</b> Connexion avec une BDD externe .....	18
<b>Figure 4:</b> La table "Info_table" de notre BDD .....	19
<b>Figure 5:</b> Les méthodes de la classe "DatabaseHelper" .....	19
<b>Figure 6:</b> Code source de la méthode "AjouterInfo" .....	20
<b>Figure 7:</b> l'interface graphique de l'application. ....	20
<b>Figure 8:</b> l'application "SqliteApp" .....	21
<b>Figure 9:</b> le MCD de la BDD .....	24
<b>Figure 10:</b> Diagramme de cas d'utilisation générale. ....	25
<b>Figure 11:</b> Les interfaces constituant notre application. ....	<b>Erreur ! Signet non défini.</b>
<b>Figure 12:</b> La clé de Google Map.....	25
<b>Figure 13:</b> Code Source de la fenêtre du marker.....	26
<b>Figure 14:</b> Les permissions ajoutées dans l'AndroidManifest.xml" .....	27
<b>Figure 15:</b> l'interface d'Accueil de l'application. ....	28
<b>Figure 16:</b> Interface graphique des "MEDECINS" .....	29
<b>Figure 17:</b> les Interfaces de Google Map. ....	30

**Liste des tableaux :**

**Tableau 1:** Les différentes versions d'Android ..... 10  
**Tableau 2:** Tableau comparatif entre Eclipse et Android Studio ..... 13  
**Tableau 3:** Comparaison entre BDD interne et externe ..... 18

## **Liste des abréviations**

OHA: Open Handest Alliance.

BDD: Base de Données.

SDK: Software Development Kit.

ADT: Android Development Tools.

ADB: Android Debug Bridge.

MCD: Modèle Conceptuel de Données.

MLD: Modèle Logique de Données.

API : Application Programming Interface.

JSON : JavaScript Object Notation.

XML : eXtensible Markup Language.

## Résumé

Durant ce modeste travail, nous nous sommes intéressés aux développements d'application mobile sous la plateforme « Android »

Notre travail a été réalisé en deux parties :

Une partie théorique dans la quel nous étudions cette plateforme en voyant son historique, ses différentes versions, son architecture et ses outils de développement ainsi que l'utilisation des bases de données sous Android.

Et une partie pratique dans laquelle nous avons réalisé une application Android, qui recherche et localise les différents corps médicaux à Tlemcen et qui donne la possibilité à l'utilisateur de les contacter via email ou téléphone.

## Abstract

During this modest work, we are interested in mobile application developments under the platform "Android"

Our work was conducted in two parts:

A theoretical part in how we study this platform saw its history, various versions, architecture and its development tools and the use of databases in Android.

And a practical part in which we have achieved an Android application that searches and locates the various medical bodies in Tlemcen and gives the possibility for the user to contact them via email or phone.

## ملخص

خلال هذا العمل المتواضع، اهتمنا في تطوير تطبيقات الهاتف الذكي تحت منصة "أندرويد".  
وقد أجريت عملنا في جزأين :

الجزء النظري في كيفية ندرس هذه المنصة شهد تاريخها، الإصدارات المختلفة، وهندستها، واستخدام قاعدة البيانات تحت أندرويد.

والجزء العملي الذي حققنا تطبيق أندرويد الذي يبحث ويحدد موقع مختلف الهيئات الطبية في تلمسان ويعطي إمكانية للمستخدم الاتصال بهم عن طريق البريد الإلكتروني أو الهاتف.