



BASE DE DONNÉES AVANCEÉ

M1, IFI 2016

Enseignant: Vu Tuyet Trinh

AndJoy

Application Android de Recherche et Localisation des endroits publics

Rapport de Projet

Étudiants: Ginel Dorleon

Fotsing Sikadie Gervais Sertilange

Hanoï, Octobre 2016

Table des Matières

I. INTRODUCTION.....	3
II. OBJECTIFS.....	4
III. GÉNÉRALITÉS.....	5
IV. LE CONCEPT ANDROID.....	5
A- Présentation de la plateforme Android.....	6
1- Architecture Android.....	6
2- Les composants d'une application Android.....	8
3- Processus de développement d'une Application sur Android.....	9
V- ANALYSE ET CONCEPTION DE L'APPLICATION.....	10
A- Analyse.....	10
B- Conception.....	11
1- Couche Accès aux données.....	12
1.a- Présentation de Google API.....	12
2- Couche Logique Métier.....	13
3- Couche Interface Utilisateur.....	14
VI- QUALITÉ DE NOTRE APPLICATION.....	17
VII- SPÉCIFICATION DE L'APPLICATION.....	18
1. Module présenté dans ce rapport.....	18
2. Modules de bases - Fonctionnalités.....	18
3. Schéma synoptique de l'architecture du système.....	19
4. Application Backload.....	20
VIII- DIAGRAMME DE CONCEPTION.....	21
1- Diagramme de Cas d'Utilisation.....	21
2- Diagramme de package.....	22
IX- IMPLÉMENTATION.....	23
1. Environnement Matériel.....	23
2. Langage, logiciel, design et environnement de programmation.....	24
3. Cas de Test et Résultat.....	24
a) Le Lancement de l'application.....	25
b) La page principale.....	25
c) Écran du Menu principal.....	25
d) Ajuster les paramètres.....	26
e) Cas de recherche de tous les Hôtel dans un rayon de 5 Km de l'endroit de l'utilisateur.....	27
f) Ajouter un endroit dans favori.....	27
X- CONCLUSION.....	28
XI- CODES SOURCES DU PROJET.....	28

I. INTRODUCTION

"Les mobiles sont aussi différents de l'internet que la télé l'a été de la radio", a annoncé Tomi Ahonen, auteur technologique et médiatique à succès. Les téléphones portables sont devenus les premiers médias de masse dans le monde (on compte 4.2 milliards d'abonnés au téléphone mobile). Il n'est pas un PC plus bête, mais un téléphone portable est considéré comme un autre support à générer des formes médiatiques.

Il a subit une évolution à une vitesse surprenante, passant du premier téléphone portable inventé par Dr Martin Cooper -un directeur général de Motorola- en 1973 qui a été la première personne à faire un appel depuis son téléphone portable, aux ordiphones, PDA et Smartphones qui disposent d'un système d'exploitation adoptant des applications tierces qui leur sont dédiées.

L'invention du premier PDA au monde, Le PenPad conçu par Apple, était dans le but de pouvoir prendre des notes, gérer son agenda, ses adresses, effectuer des calculs, etc, sans avoir à s'encombrer d'un ordinateur portable ou d'un bloc notes.

Aujourd'hui ces périphériques ont atteint une puissance de calcul énorme, une taille mémoire ainsi qu'un débit nécessaire pour faire tourner des applications aussi diverses que variées qui vont de l'Outlook mobile jusqu'aux applications de navigation GPS.

Les plates-formes de distribution de ces applications sont en plein essor, Windows à son magasin d'applications, l'Apple à l'App Store, Android à son Market, etc.

Ce qui ne cesse d'inciter beaucoup de développeurs à l'élaboration des applications très prisées en profitant des multiples apports des plateformes présentes sur le marché et des diverses innovations technologiques (Wifi, GPS, GPRS, etc.).

En effet, la géolocalisation à l'aide du GPS des « téléphones intelligents » est très utile aux applications comme les annuaires, portails et autres outils permettant de trouver ce que l'on cherche autour d'un lieu, répondant par la fin aux besoins quotidiens des utilisateurs. Les informations sont disponibles, les besoins et les modes d'accès sont ceux qui hantent les utilisateurs. Il est toujours du coté des utilisateurs de faire le choix des moyens les plus faciles d'accéder aux informations voulues

C'est dans ce cadre que s'inscrit notre projet, une application Android, pour le cours de base de données avancées. Intitulée « AndJoy », l'objectif est de concevoir une

application mobile dédiée au téléphone de la plateforme Android, permettant à un utilisateur de localiser les endroits autour de lui grâce aux outils et techniques de géolocalisation et d'orientation offerts par cette plateforme.

Pour ce faire, nous procémons par une étude théorique afin de mieux cerner le contexte de notre travail

II. OBJECTIFS

Notre application Android AndJoy est conçue pour être une suite complète comprenant la géolocalisation de différents types d'endroits: ATM, Banques, Hôtel, Hôpitaux, Resto tout en offrant la route et les distances pour accéder à ces endroits. Il existe beaucoup d'applications payantes en ligne permettant la localisation des endroits publics mais cependant l'accès à internet est une condition requise pour en accéder, alors nous proposons une solution libre, gratuite, performante et disponible hors-ligne et en ligne adaptée aux différents modèles de périphériques tournant sur Android permettant aux utilisateurs de répondre à leur besoin de localisation. Notre application est implémentée suivant l'architecture client-serveur et accessible sur tout dispositif Android.

Afin de se rendre à un lieu, l'utilisateur tape ou choisit l'endroit qui l'intéresse. Par la suite, le terminal envoie une requête au serveur qui effectue les traitements nécessaires et renvoie la liste des endroits les plus appropriés suivant le rayon défini. Les informations reçues du serveur sont affichées sur une carte au niveau du terminal et les chemins à parcourir. Durant un déplacement, l'utilisateur est guidé par le biais d'un pointeur indiquant sa position courante sur une carte ainsi que d'autres indicatifs de type tournez à droite, à gauche, etc. Une fois qu'il arrive à la fin du trajet, l'utilisateur est alerté.

Cette application sera librement distribuée sur le playstore de Google.

III. GÉNÉRALITÉS

Étant donné que nous suivions une discipline utilisant la notion de base de données avancées , en plus du nombre grandissant d'applications tournant sur mobile et utilisant des données géographiques et spatiales, nous avons proposé de développer cette application sur cette nouvelle plateforme pour appareils mobiles : Android.

Notre défi était donc de réaliser un projet utile, lié à l'actualité et qui satisfait les critères et les objectifs du cours de base de données avancées. Il nous a fallu trouver un sujet qui nous permettra d'utiliser un maximum de concepts de cette nouvelle plateforme. Ce projet est purement pédagogique et nous permet de comprendre l'importance pour un informaticien de suivre l'actualité informatique ainsi que de se confronter régulièrement à de nouvelles technologies.

Afin de comprendre comment réaliser une telle application, il nous faut présenter, tout d'abord l'évolution de la plateforme Google Android. Nous présenterons la plateforme Google Android, ses mécanismes ainsi que les outils qui nous permettent de la manipuler. Ensuite, nous détaillerons notre méthode pour répondre au problème aussi bien au niveau conceptuel que techniques. Et enfin nous pourrons présenter nos résultats ainsi que nos réflexions sur l'avenir de ce projet.

IV. LE CONCEPT ANDROID

Android a été développé par une startup de même nom, racheté par la suite par la société Google. Mais depuis sa première sortie en 2007, il est développé au sein de *l'Open Handset Alliance*, qui regroupe une trentaine de partenaires. En 2008, Android est devenu une plateforme en code source ouvert.

Android est basé sur un noyau linux (ce qui lui confère un très grand avantage sur la sécurité et la puissance des systèmes linux), et comporte une plateforme développée en Java, dans lequel les applications sont exécutées via un interpréteur JIT. C'est un système gratuit et complètement ouvert, car le code source et les APIs sont ouverts. Ainsi les développeurs ont la permission d'intégrer, d'agrandir et de remplacer les composants existants. C'est aussi un système très flexible. C'est-à-dire qu'un constructeur peut adapter les applications aux besoins du client ou remplacer entièrement. C'est l'un des avantages qui a permis aux constructeurs de maîtriser leur produit, ainsi d'apporter une valeur ajoutée à leur téléphone, pour ne pas être un simple assebleur comme pour les ordinateurs.

En termes d'application, Android a été conçu pour intégrer plusieurs services Google comme Gmail, Google Maps, Google Agenda, Google Talk, YouTube. Avec un accent particulier sur la géo localisation avec Google Latitude et la météo.

Pour développer une application sur Android, Google a mis à la disposition de la communauté Open Source un SDK, qui offre en complément les APIs.

En fin des années 2010, Android était devenu un concurrent sérieux face aux autres plateformes, car le nombre de mobile tournant sur celui-ci était en grande expansion.

A- Présentation de la plateforme Android

Cette partie présente Android du point de vue de l'intérieur, en explorant les aspects techniques internes. Exploration nécessaire pour tout futur développement la plateforme.

1- Architecture d'Android

Système d'exploitation Open Source pour terminaux mobiles conçu par le Startup Android rachetée par Google en Août 2005, Android est aujourd'hui développé autour du consortium Open Handset Alliance réunissant une trentaine de partenaires. Cet OS se différencie des autres systèmes pour mobile parce qu'il possède un avantage majeur le fait qu'il est ouvert.

C'est pourquoi, il nous a semblé pertinent dans le cadre de ce projet académique de travailler sur cette plateforme.

Les applications Android sont développées en JAVA. Android dispose d'un ensemble de librairies du JDK, en plus des fonctionnalités supplémentaires qui lui sont propres (c'est le cas par exemple de la gestion de l'interface graphique). Étant en JAVA, on aurait pu penser que ces applications s'exécutent sur la JVM (machine virtuelle java de Sun). Ce qui n'est pas le cas, car elles s'exécutent sur la "**Dalvik VM**".

Il s'agit de la machine virtuelle développée par Google pour Android et adaptée aux systèmes restreints en mémoire et puissance de processeur.

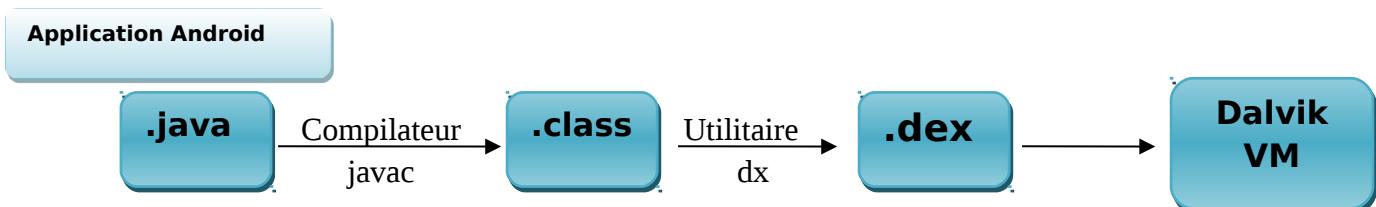


Figure i: Conversion d'un fichier .java en un fichier .dex exécutable sur la machine virtuelle Dalvik.

La figure de la page précédente montre comment se fait le processus de transformation, avec tout d'abord la compilation d'un fichier .java en .class, puis sa conversion au préalable au format dex avec l'outil dx.

Une autre particularité de cette plateforme est que chaque application s'exécute dans sa propre instance de la Dalvik VM, le système étant multitâche, il doit être capable d'exécuter en même temps plusieurs instances de la VM.

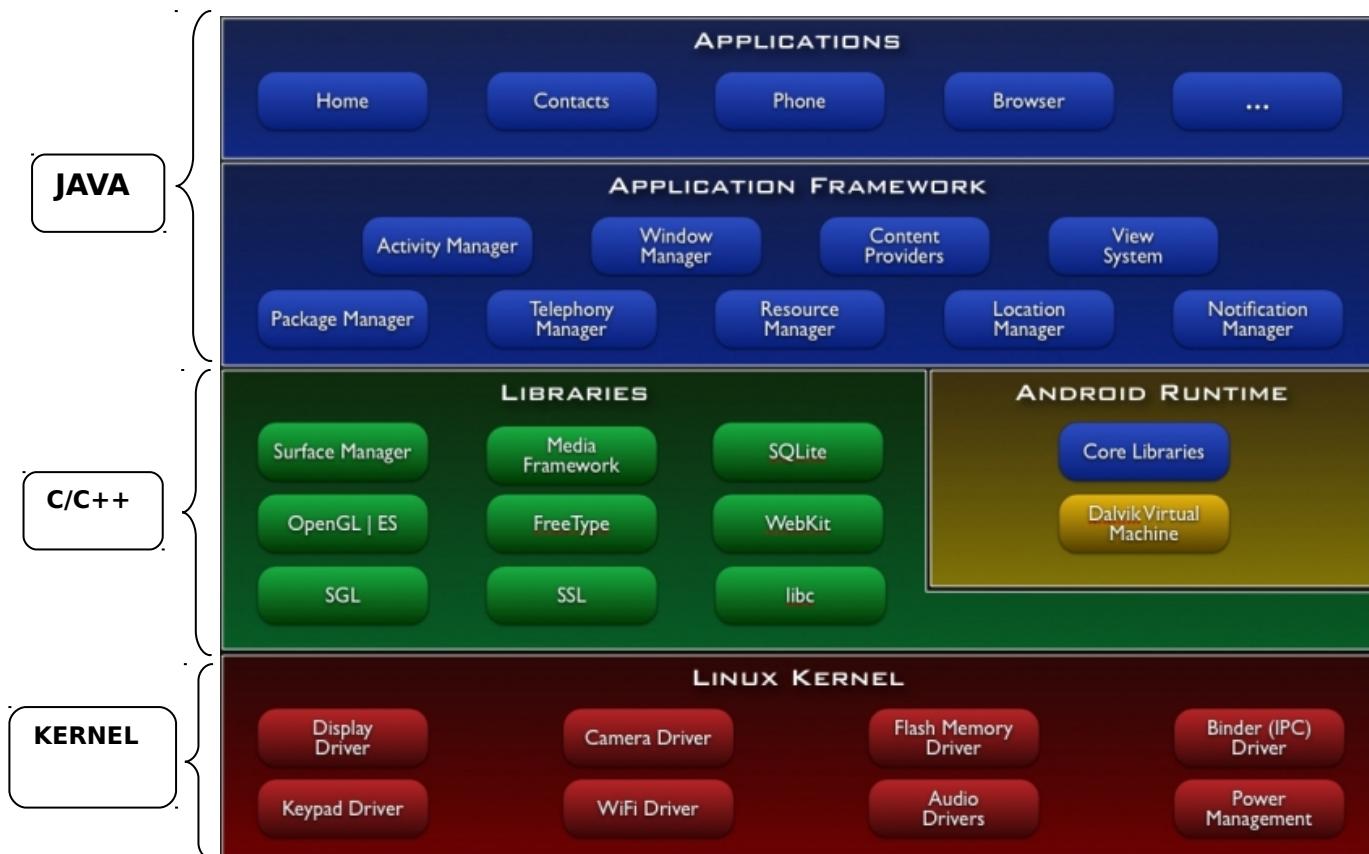


Figure ii: Architetture du systeme android

En observant la figure ci-dessus, on constate qu'Android est basé sur le noyau Linux 2.6.XX. Alors il bénéficie de plusieurs avantages dont dispose le système Linux, tels qu'une bonne gestion de la mémoire et des processus, modèle de sécurité, soutient de bibliothèque partagée, etc.

Au dessus, on trouve une collection de plusieurs bibliothèques C/C++, utilisées par un certain nombre de composant du système Android.

Au dessus de ceux-ci, on trouve l'Android Runtime, siège de la Dalvik VM et les librairies coeurs du Framework.

Plus au dessus, on a la couche "Application Framework", couche permettant au développeur de créer des applications.

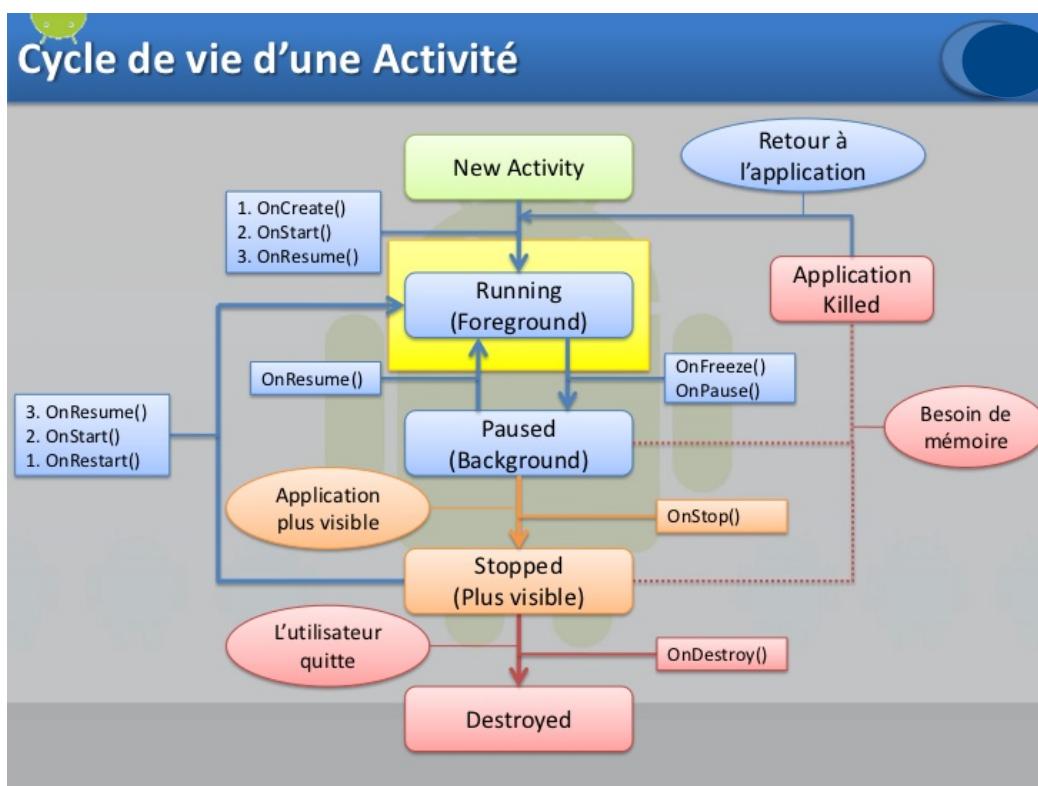
Enfin on trouve les applications au dessus du Framework. Ce sont les applications qui marchent sous la plateforme Android. Toutes développées en Java.

2- Les composants d'une application Android

Les applications Android sont composées de 4 principaux composants :

- ❖ Activity : Gestion de la vie des applications, car une application se compose de plusieurs écrans. Et chaque écran peut être réalisé par une activité, implémentée sous la forme d'une classe qui hérite de la classe Activity.
- ❖ Services : Les services n'ont pas d'interface graphique et tournent en tache de fond (en arrière-plan). C'est-à-dire qu'une application s'exécute quand une autre application est en train de s'exécuter comme les services de lecture de musique.
- ❖ Broadcast receives : Il écoute et réagi aux annonces broadcast (réactions sur les événements extérieurs). Par exemple appel entrant, changement de fuseau horaire.
- ❖ Content providers : Gestion du partage de données entre applications.

En résumé, Android permet le partage de composant entre applications ainsi que de gérer leur cycle de vie, le cycle de vie des activités plus précisément. Ce cycle de vie est exprimé par la figure ci-dessous.



3- Processus de développement d'une Application sur Android

Le processus de développement d'une application sur Android, suit le processus de développement d'une application classique à des différences près. Des différences qui sont liées au fait que l'application doit tourner sur des terminaux mobiles.

- Conception de la base de données
- Création des classes pour représenter les données physiques (couche Mapping). Afin de définir des actions comme : supprimer, ajouter, modifier.
- Dessiner des interfaces en fichiers XML ou en codage
 - Les vues (View),
 - Les arrangements (Layout).
- Choisir les arrangements. On indique les interfaces des activités. Android définit deux techniques pour exprimer les interfaces. Par fichiers XML ou en programmant directement les composants dans le code.
- Organiser les ressources de l'application : Icônes, images, chaînes (dans le fichier String.xml), etc.
- Créer et mettre à jour le fichier de configuration AndroidManifest.xml, il est important de laisser la plateforme le faire.
- Créer les activités :
 - Comme énoncé plus haut, chaque activité peut correspondre à un écran ou une fonction de l'application
 - Celles s'occupant des méthodes pour communiquer avec la base de données
 - Les Intents (Instanciations de classes) sont utilisées pour orienter les activités.

Ainsi, nous venons de présenter la plateforme Android et le processus de développement sur celle-ci, il est maintenant temps de voir la configuration de l'environnement nécessaire au développement de notre application.

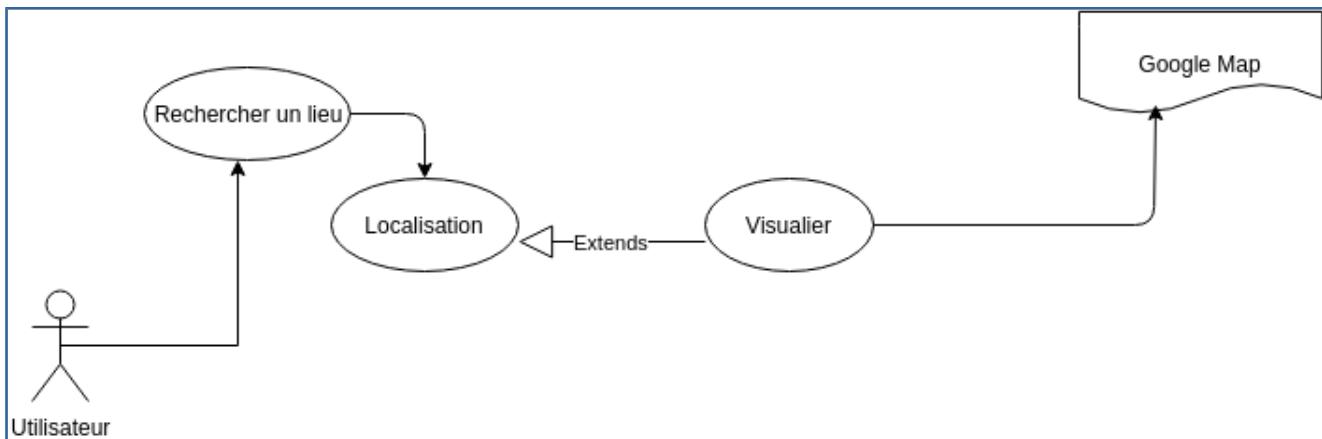
V- ANALYSE ET CONCEPTION DE L'APPLICATION

Ayant pris connaissance de la plateforme Android et aux spécificités liées à celle-ci, il est maintenant utile de faire une analyse de la conception de notre application.

A- Analyse

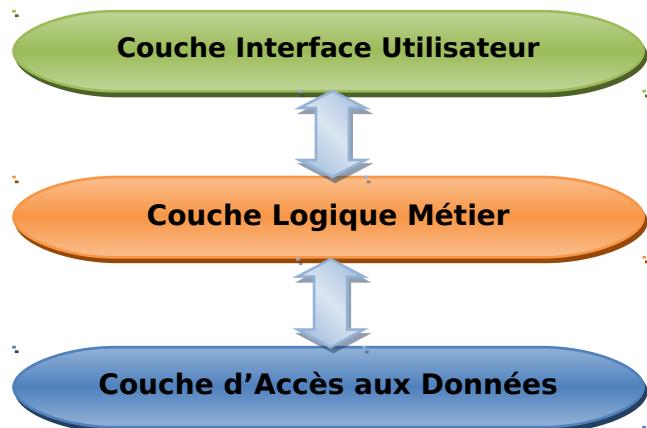
Comme tout début d'un projet informatique, nous avons réfléchi aux différentes fonctionnalités attendues par les futurs utilisateurs de l'application. Nous sommes limités au but principal de l'application : Application Android permettant de faire la recherche et la localisation des endroits publics d'un espace géographique.

L'utilisateur doit être capable d'obtenir la localisation (longitude et latitude). Cette localisation est donnée par le GPS du téléphone de l'utilisateur, et en utilisant l'API Google Maps, l'application fera une visualisation des coordonnées fournies par ce GPS sur la carte.



B- Conception

La technologie actuelle du Génie Logiciel impose la prise en compte d'un concept important, le plus important même, qui est celui de la ré-utilisabilité des composants. Cela passe donc par la prise en compte de la notion de conception en couche ou n-tiers, dès la phase de conception des logiciels. Ainsi afin de respecter ce principe pour permettre une bonne gestion de la qualité de l'application, nous avons utilisé une conception 3-tiers. Ce découpage est présenté de manière schématique par la figure suivante :



Afin de suivre et de mieux comprendre le processus développement, nous allons tour à tour procéder à un développement de bas en haut, de la couche d'accès aux données vers la couche interface utilisateur.

1- Couche Accès aux données

La couche d'Accès aux données de notre application repose essentiellement sur les données géographiques acquises par le GPS. Ainsi, pour accéder à ces données, nous avons utilisé les services de Google Maps API.

Google Places API est le Service Web de Google qui renvoie des informations sur les lieux (définis dans cette API en tant qu'établissements, points géographiques ou points d'intérêt importants) à l'aide de requêtes HTTP.

1.a- Présentation de Google API

Google API est un ensemble d'interfaces de programmation d'applications (API) développés par Google, qui permettent la communication avec les services Google et leur intégration à d'autres services. Utilisant l'API, les requêtes de lieu suivantes sont disponibles :

- Les **recherches de lieux** renvoient une liste de lieux en fonction de la position géographique de l'utilisateur ou d'une chaîne de recherche.
- Les requêtes de détails de lieu, «**Place Details** » renvoient des informations détaillées sur un lieu spécifique, notamment des avis d'utilisateurs.
- **Place Add** vous permet de compléter les données de la base de données Google Places avec les données de votre propre application.
- **Place Photos** vous donne accès à des millions de photos de lieu stockées dans la base de données Google Place.
- **Place Autocomplete** permet de renseigner automatiquement le nom et/ou l'adresse d'un lieu au fur et à mesure de la saisie.
- **Query Autocomplete** permet de fournir un service de prédiction de requête pour les recherches géographiques textuelles, en suggérant des requêtes au fur et à mesure de la saisie.

Chacun de ces services de saisie semi-automatique est accessible sous la forme d'une requête HTTP et renvoie une réponse JSON ou XML. Toutes les requêtes envoyées à un service Places doivent utiliser le protocole https:// et inclure une clé d'API.

2- Couche Logique Métier

Cette couche représente l'ensemble du métier de notre application. En suivant les étapes de la circulation des données dans une architecture 3-tiers, elle devra recevoir les données venant de la couche d'accès aux données, pour effectuer les traitements attendus par l'utilisateur.

Ainsi, la première chose effectuée par le métier pour exploiter les données émanant de la couche données est de définir le pont d'accès entre les deux niveaux. Cela est fait par les éléments suivants.

- ❖ La balise de permission d'accès au service GPS

```
<!-- Permissions -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

- ❖ Le choix du fournisseur d'accès aux informations de localisation

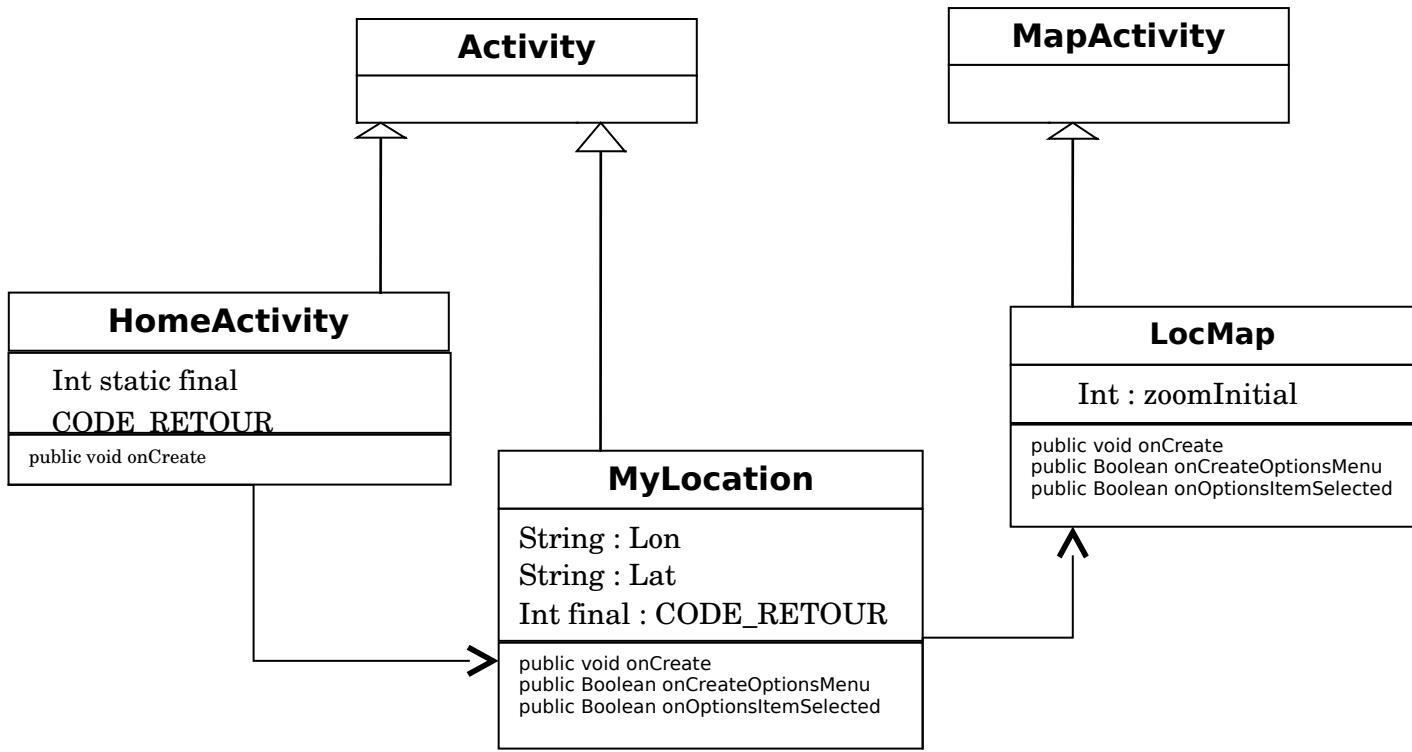
```
this.locMngr = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

Notre application de géolocalisation étant une application de recherche et de localisation, comme l'a montré le diagramme de cas d'utilisation précédent, les traitements métiers sont simples.

En effet notre activité principale (HomeActivity) hérite de la classe Activity, puis implémente la méthode onCreate qui sera appelée au démarrage de l'application. Par la suite elle instancie une autre activité (MyLocation) qui va prendre en charge l'affiche des données fournies par la couche de données.

En ce qui concerne l'affichage de la localisation sur une carte, elle est effectuée par l'activité LocMap. Activité instanciée par l'activité MyLocation. Il est important de rappeler que dans la plateforme Android l'instanciation d'une classe est appelée Intent.

En utilisant une analyse orientée objet, on déduit le diagramme de conception suivant. Il résume de manière fonctionnelle le fonctionnement de la couche métier de cette application de géo localisation.



3- Couche Interface Utilisateur

Cette couche est composée de trois interfaces qui servent d'interaction entre l'utilisateur et la couche métier. Une de ces interfaces est celle qui permet d'afficher la localisation géographique.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:id="@+id/head"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10sp"
        android:layout_marginBottom="12sp"
        android:text="Current Location"
        android:textColor="#0f0"
        android:textSize="20sp">
  
```

```

</TextView>

<TextView android:id="@+id/TexteLat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Latitude : "
    android:layout_below="@+id/head"
    android:layout_marginBottom="12sp"
    android:textStyle="bold"
    android:layout_marginTop="16sp">
</TextView>

```

En effet, la plateforme Android propose deux types d'alternatives pour la création des interfaces.

- Sous forme de fichier XML. Ces fichiers sont placés dans le dossier **/res/layout** du répertoire du projet. Cette méthode permet d'alléger le code afin que l'on ne concentre que sur la logique de l'application (réactions de l'application aux actions de l'utilisateur)
- Directement dans le code. Très peu utilisé par les développeurs Android, car elle augmente la densité du code.

Comme on vient de le voir, il est nécessaire de définir une logique pour l'utilisation des interfaces, elle va établir le lien entre les interfaces utilisateurs et la couche métier.

Dans le cas de notre application, attardons nous un peu sur la logique qui se trouve autour de l'interface ci-dessous. En effet tout part de la première Activité de l'application. La première méthode à être appelée est `onCreate`, qui charge l'interface de la 1^{ère} Activité par le biais de **R.layout.main**. Lorsque l'utilisateur clique sur le bouton de localisation, la fonction (GPS) les coordonnées géographiques de l'utilisateur.

```
loc = HomeActivity.this.locMngr.getLastKnownLocation("gps");
```

Puis cette Activité instancie la 2e Activité, car c'est celle-ci qui va se charger d'afficher les données de localisation récupérées.

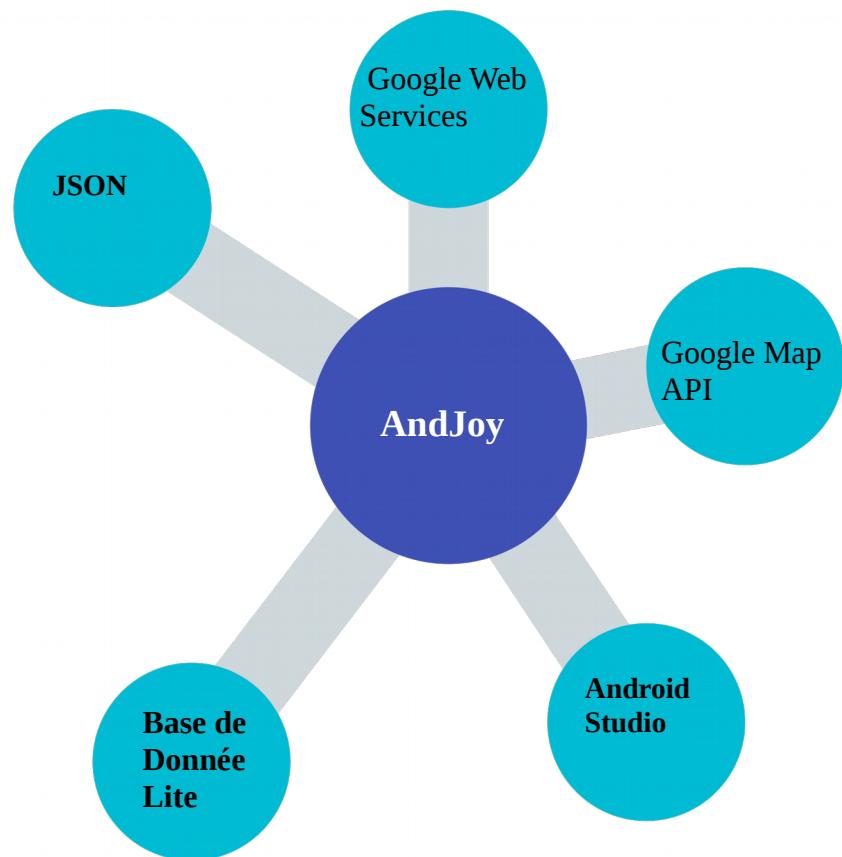
```

Intent intent;
intent = new Intent(HomeActivity.this, MyLocation.class);
Bundle bundle = new Bundle();
bundle.putDouble("Latitude", loc.getLatitude());
bundle.putDouble("Longitude", loc.getLongitude());
intent.putExtras(bundle);
startActivityForResult(intent, CODE_RETUR);

```

Présentation de l'Application.

En général, la structure de notre application utilise les services suivants :



VI- QUALITÉ DE NOTRE APPLICATION

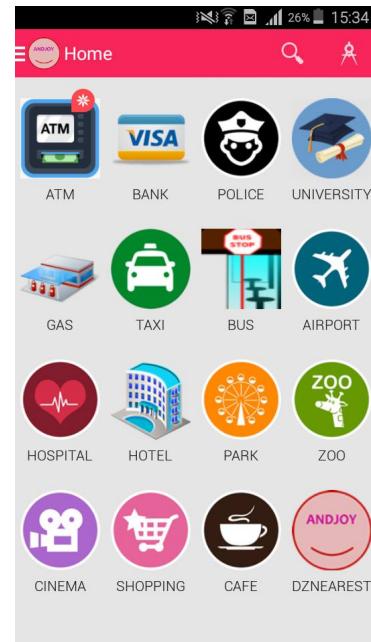
- Ergonomique :L'interface utilisateur représentant l'ensemble de l'environnement de l'application est très adaptée à la vue et est facile à comprendre.
- Portabilité : N'importe quel matériel équipé du noyau d'une version Android supérieure à 1.1 peut exécuter notre application.
- Gratuité : L'application est gratuite sur le store de Google.
- Évolutivité: Possibilité de mises à jour
- Adaptabilité: Le logiciel peut être paramétré et adapté à divers types de modèle de téléphones.
 - ◆ Installation sur tout matériel supportant Android.
 - ◆ Accès aux données en local ou via internet.
 - ◆ Interface dynamique, ergonomique et modulable qui s'adapte à la taille de l'écran de n'importe quel matériel.
 - ◆ Une personnalisation simple et complète en fonction du type de recherche
 - ◆ Une présentation très 'visuelle' des données
 - ◆ Une prise en charge simultanée des différentes critères de recherche

VII- SPÉCIFICATION DE L'APPLICATION

1. Module présenté dans ce rapport

Dans ce rapport et pour cette première version (*version 1.0*) de AndJoy, nous présentons un module de recherche sur 15 endroits publics:

1. ATM
2. BANK
3. POLICE STATION
4. UNIVERSITY
5. GAS
6. TAXI
7. BUS STATION
8. AIRPORT
9. HOSPITAL CENTER
10. HOTEL
11. PARK
12. ZOO
13. CINEMA
14. SHOPPING
15. CAFE



2. Modules de bases - Fonctionnalités

Concrètement, notre application Android permet à un utilisateur de faire la recherche sur un endroit bien défini. Avec AndJoy, vous pouvez facilement trouver à proximité de vous et dans un rayon défini des ATM, Banques, Hôtel, Hôpitaux,Café, Taxi, Station de Bus, Poste de Police, Université, Cinéma etc.. ou tout autre endroit que vous désirez chercher.

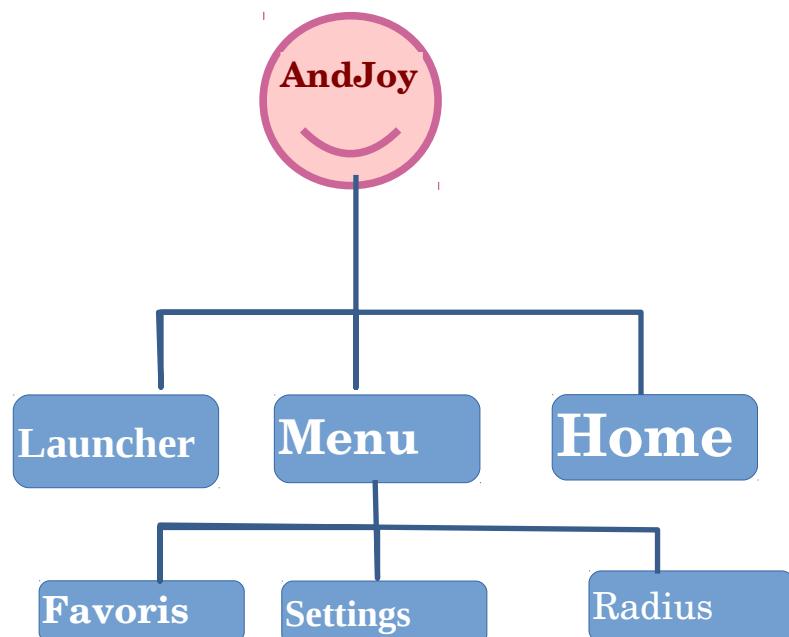
AndJoy offre les fonctionnalités suivantes:

- Chaque endroit recherché est fixé avec tous les détails, y compris l'adresse de l'endroit, les images, le site internet, notes , numéro de téléphone, la direction pour atteindre l'endroit, avis des utilisateurs et la distance de l'endroit d'où vous

vous êtes actuellement.

- Un grand outil plan de direction avec le temps et la distance approximative est disponible pour vous diriger comment atteindre la destination sélectionnée.
- Les résultats peuvent être triés par la distance ou de la note des internautes.
- Tous les résultats de la recherche peuvent être localisés sur le map de Google.
- Vous pouvez ajouter des emplacements à votre liste de favoris dans «My Places»
- Vous pouvez vous localiser vous-même avec l'option « My Location » du menu principal
- Prise en charge de tout type d'informations sur l'endroit recherché
- Sauvegarde automatique des données sur le téléphone dans une base de données « Lite » pendant la durée de la session.

3. Schéma synoptique de l'architecture du système



4. Application Backload

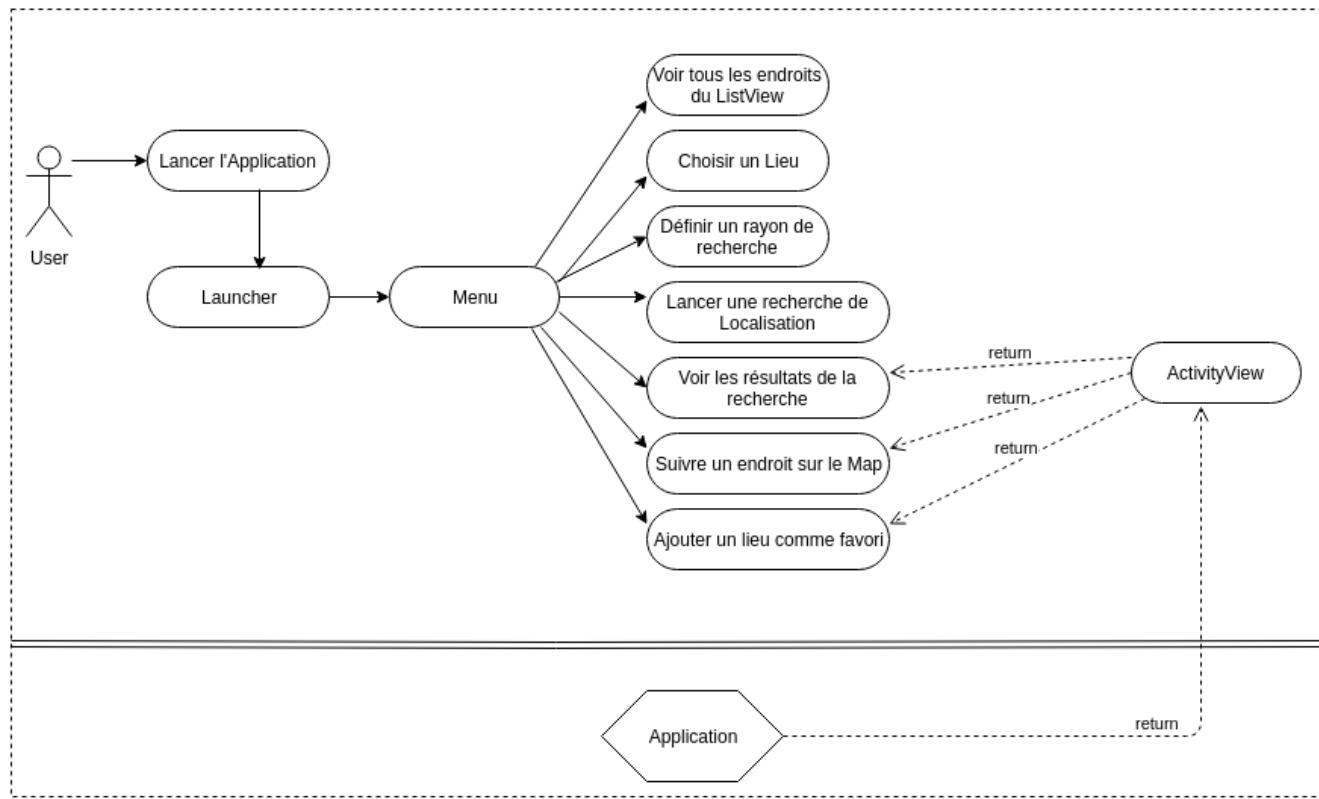
Action #	Name	How to ?	
1	Rechercher un lieu	Open the App, reach out the menu, and then tap location's icon	
2	Ajouter un lieu comme favori	Go to Home, tap a location, then click add to My Places to save it as favourite	
3	Supprimer un favori	Go to Menu, choose My Places , then choose the place you want to delete	
4	Définir un rayon de recherche	Go to Home, Click Radius to set up a radius for your search	
5	Définir critères d'affichage	Go to Menu of the app, click on Adjust, then adjust your settings.	
6	Voir la dernière recherche effectuée.	Go to Home, click on DZNearest(Drop Zone Nearest) to see your last research.	

VIII- DIAGRAMME DE CONCEPTION

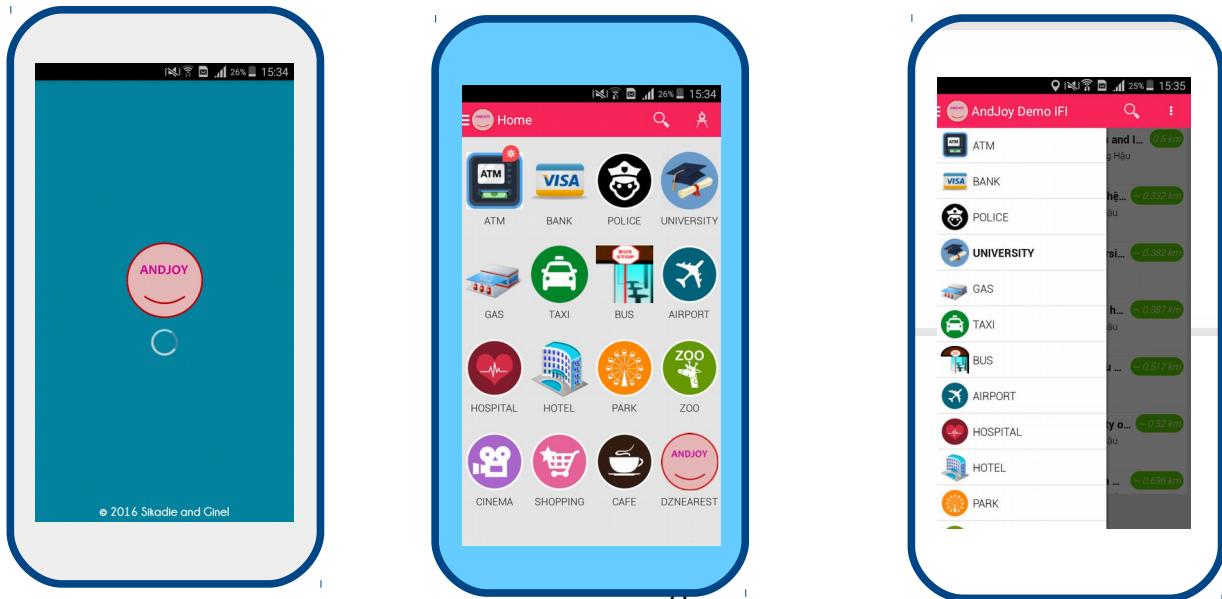
1- Diagramme de Cas d'Utilisation

Les diagrammes de cas d'utilisation représentent généralement toutes les interactions des utilisateurs avec le système.

Dans le schéma suivant, nous présentons le cas-d'utilisation incluant le lancement de l'application et l'accès au menu et des différentes options.



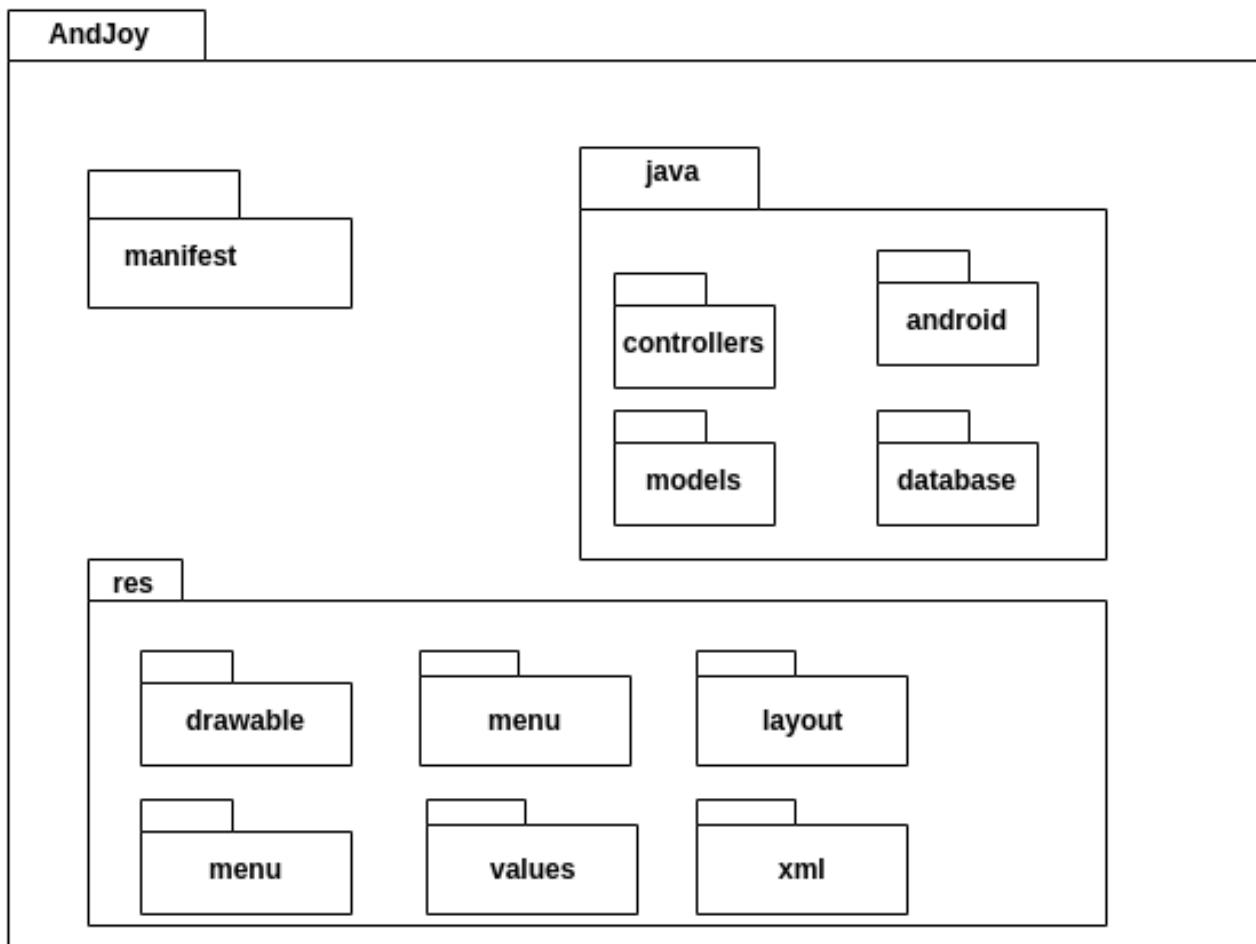
Capture d'écrans illustrant ce diagramme



2. Diagramme de package

Le diagramme de paquetages est la représentation graphique des relations existant entre les différents paquetages (ou espaces de noms) composant un système.

Ici, on peut voir le diagramme de tous les packages du système.



Le package manifest contient la classe AndroidManifest.xml

Le package java contient toutes les classes java, les controllers, les activités

Le package res, il contient toutes les ressources principales du système, les parseurs etc..

IX- IMPLÉMENTATION

1. Environnement Matériel

Le développement du système a été réalisé sur une machine HP exécutant le système d'exploitation Linux, Ubuntu version 16.04

Un processeur AMD E1-2100 APU with Radeon(TM) HD Graphics × 2

Une carte graphique Gallium 0.4 on AMD KABINI (DRM 2.43.0, LLVM 3.8.0)

Une mémoire vive de 4Go.Un disque dur 500 Go.

Un écran 14 pouces.

2. Langage, logiciel, design et environnement de programmation

JAVA: Langage de programmation

AndroidStudio :IDE

Android : SDK 23

JDK 1.8 +

UML :Modélisation

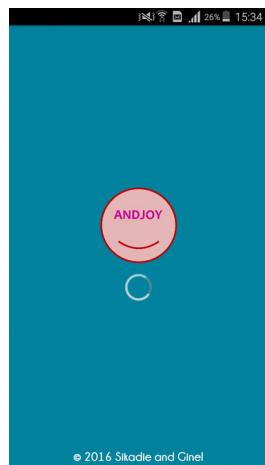
JSon

3. Cas de Test et Résultat

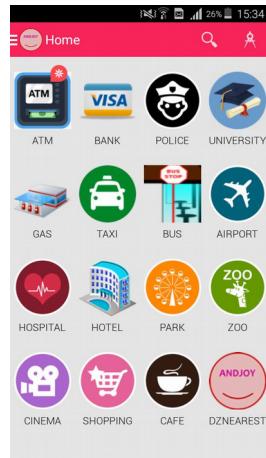
Selon l'IEEE (Standard Glossary of Software Engineering Terminology), le test est l'exécution ou l'évaluation d'un système ou d'un composant par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus.

Dans les images ci-dessous, nous présentons les résultats des différents cas de test manuellement effectués.

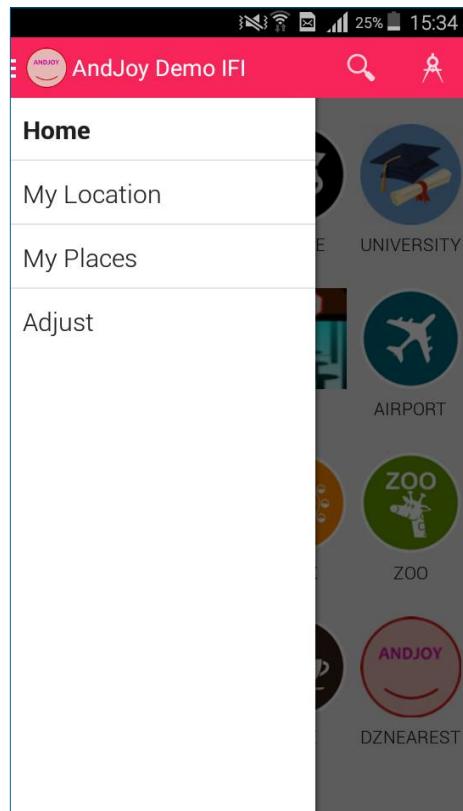
a) Le Lancement de l'application



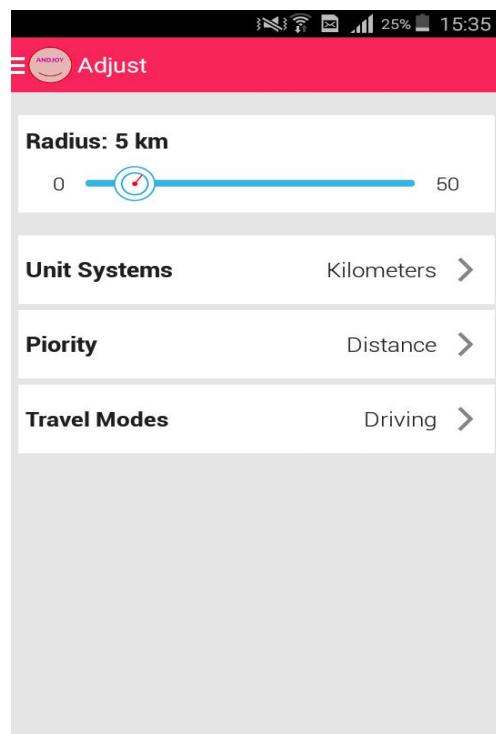
b) La page principale



c) Écran du Menu principal



d) Ajuster les paramètres



La capture d'écran ci-dessus représente les paramètres pour la configuration des critères de recherche.

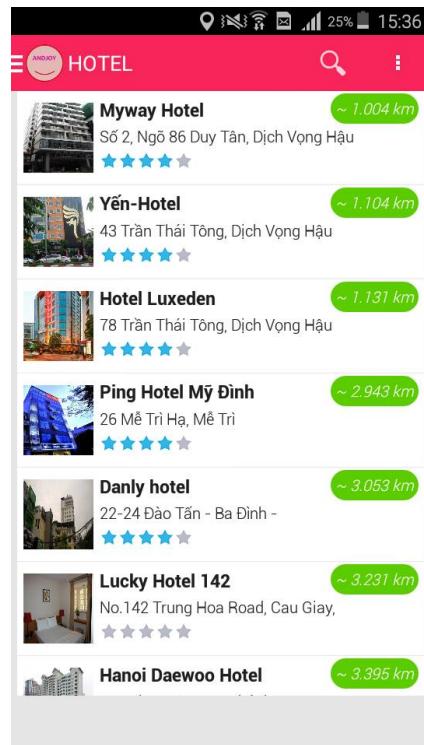
Radius: permet de définir la distance ou le rayon de recherche.

Unit System: pour définir l'unité des distances, soit Kilometers, soit Miles

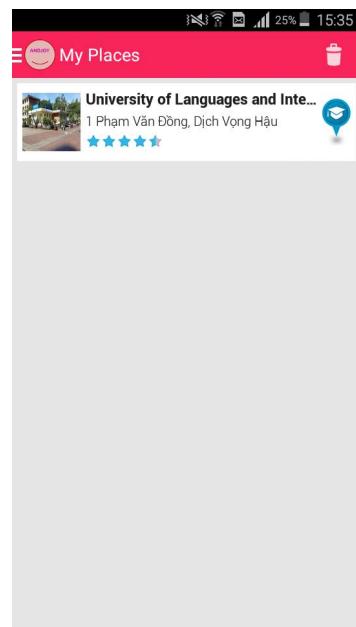
Priority: Permet de trier et d'afficher les résultats soit par la priorité des distance ou soit par les avis des utilisateurs. Il peut être soit Distance soit Rating

Travel Modes: Permet de définir le mode de déplacement pour atteindre un endroit. Il peut être soit Driving ou Walking

e) Cas de recherche de tous les Hôtel dans un rayon de 5 Km de l'endroit de l'utilisateur



f) Ajouter un endroit dans favori



X- CONCLUSION

En effet, le but réel de ce projet de Base de Données Avancées était de voir l'aptitude de chaque étudiant à réaliser un projet avec les nouvelles connaissances apprises dans le cadre du cours.

Nous pouvons parler d'un bilan positif sur plusieurs plans. En effet, sur le plan pratique, la bonne compréhension de la plateforme Android, ayant déjà une bonne base Java. La bonne maîtrise des services de géolocalisation, l'utilisation des services web et de l'api de Google Map, ainsi tout le déploiement de notre application Android exploite tous les services de Google.

En ce qui concerne l'objectif pédagogique, nous sommes particulièrement satisfaits avec les nouvelles notions de bases de données spatiales, l'utilisation des points, des polygones et des lignes permettant de localiser dans le temps et dans l'espace les endroits autour de nous. C'est une nouvelle connaissance et c'est important pour un informaticien.

En termes de difficultés, elles étaient énormes, la signature de l'application Android, la création de la clef de signature, la récupération des données et la redirection de ces données sur les serveurs de Google étaient en effet des difficultés que nous avions rencontrées. Mais avec l'aide des forums et de nos recherches nous avions pu les surmonter et produire un résultat satisfaisant.

En bref, ce projet nous a permis d'acquérir désormais des compétences dans le développement sur la plateforme Android. Notre application AndJoy est à sa première version, mais nous comptons faire des mises à jour suite aux commentaires des utilisateurs.

XI- CODES SOURCES DU PROJET

Consulter les codes sources du projet en développement sur le répertoire en ligne de Github :

<https://github.com/gdorleon/AndJoy>

Références Bibliographiques

- [1] <https://developers.google.com/places/web-service/intro>
- [2] <https://developer.android.com/guide/topics/location/strategies.html>