

Gestion GPS en fonction de la localisation à l'aide du SDK Android

Plan :

Ce projet va vous être présenté en différentes parties :

- I. Introduction
 - 1. Déroulement du développement
 - 2. Tutoriel Android, « créer son premier projet Android »
- II. Conception du projet
 - 1. Schématisation UML
 - 2. Technologies utilisées
- III. Analyse Technique
 - 1. Approche globale
 - 2. Problématique base de données
 - 3. Problématique API Google Maps
- IV. Mise en œuvre
 - 1. Diagramme de cas d'utilisation
 - 2. Implémentation, description des différentes classes.

I. Introduction :

1. Déroulement du développement

Le but de cette application est de permettre à des utilisateurs Android de localiser les bars dans les alentours de la localisation GPS de l'utilisateur.

Le développement de cette application s'est déroulée en plusieurs étapes.

- Etude des fonctionnalités de l'application
- Etude de l'environnement de développement Android ;
- Choix du SGBD, en l'occurrence mySQL ;
- Recherche des différents tutoriels nécessaires à la réalisation de l'application ;
- Conception de la BDD
- Réalisation de l'application

Afin de vous transmettre au minimum les connaissances que nous avons engendrés, nous vous proposons un tutoriel Android. Tutoriel nécessaire à la création d'un premier projet Android.

2. Tutoriel Android « créer son premier projet Android »

Ce tutoriel va porter sur l'installation de l'environnement de développement nécessaire à Android. Ce tutoriel va se décomposer en 7 étapes majeures.

Téléchargement d'Eclipse

Se rendre sur la page <http://www.eclipse.org/downloads/>
Choisi la version d'Eclipse : Eclipse Classic 3.7.2

L'installation d'éclipse consistera en la décompression de l'archive : eclipse-SDK-3.7.2-win32-x86_64.
Il suffit tout simplement de d'extraire le fichier eclipse vers l'endroit de notre choix.
On peut le mettre soit à la racine C:\eclipse soit dans nos Documents

Téléchargement du SDK Android

Se rendre sur la page : <http://developer.android.com/sdk/index.html>

On choisit la plateforme qui convient, le package [installer_r18-windows.exe](#) est recommandé car il est rapide et facile à l'installation.

Pour avoir des informations complémentaires sur l'environnement SDK, ce site permet de répondre à de nombreuses questions : <http://developer.android.com/index.html>

Les deux téléchargements sont finis passons à l'installation !

Pour le SDK Android, ce n'est qu'une simple installation, on peut choisir un autre dossier de destination que celui choisi par défaut pour l'installation des fichiers.
Tous les fichiers sont installés dans ce répertoire (par défaut) : C:\Program Files (x86)\Android\android-sdk.

Il est préférable d'installer tous les fichiers sous ce répertoire :

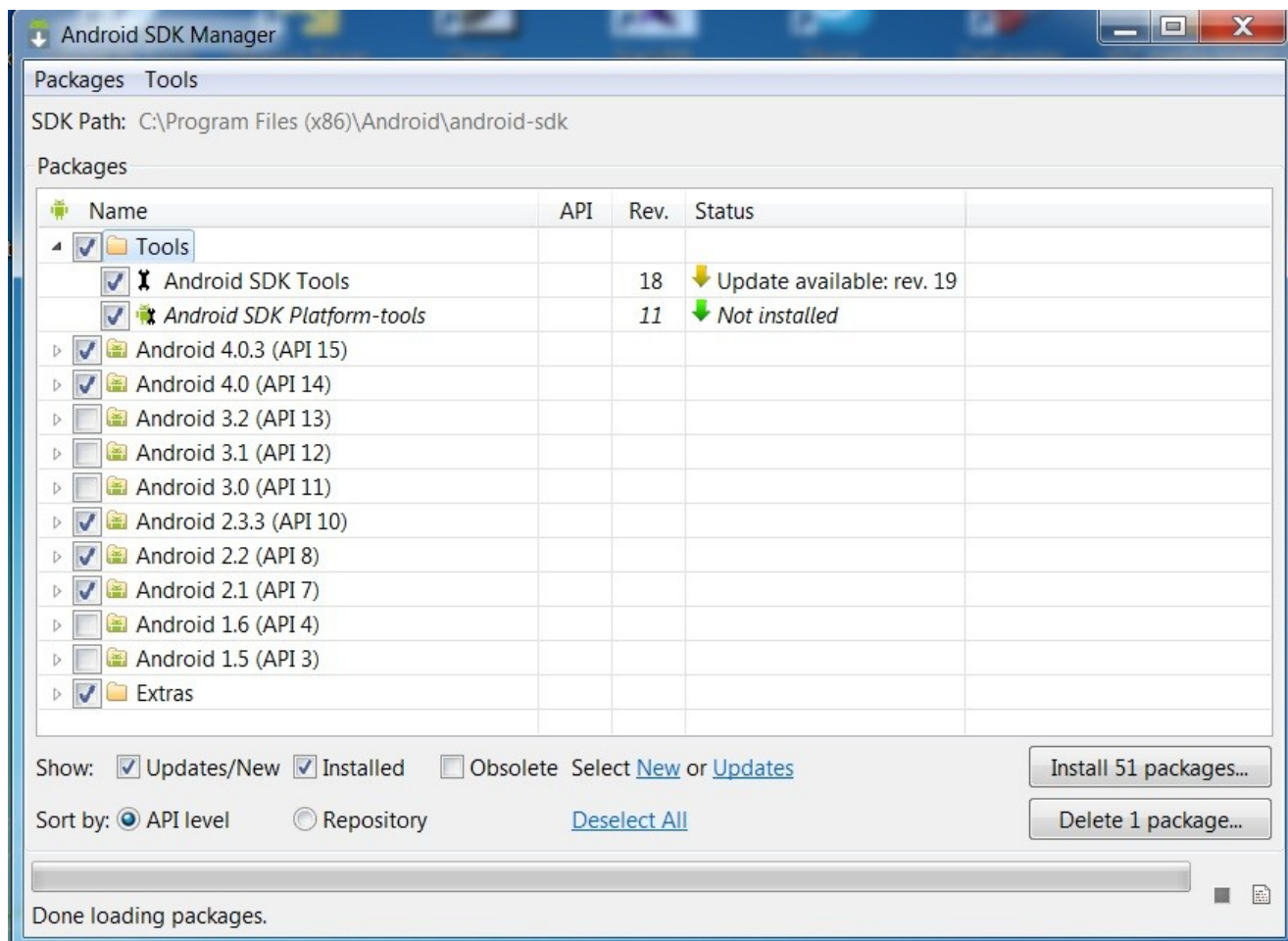
C:\Android\android-sdk

On accède aux différents packages Android : il y a les plateformes des différentes versions d'Android,

des outils, des exemples etc.

On sélectionne les différents package que l'on souhaite. Les packages nommées Android X.X sont des numéros de version du système d'exploitation. Le téléchargement des versions est assez long.

Dans cet exemple je n'ai pris que 5 versions d'Android. Ce sont les plus utilisées sur le marché, cependant il est préférable de choisir le plus de versions afin d'être compatible avec le plus grand nombre d'utilisateurs.



Une fois les différents packages installés, le statut des packages passent en « Installed »

Dernière étape, s'assurer que la machine virtuelle Android est fonctionnelle (AVD : Android Virtual Device).

Configuration d'Eclipse

Eclipse vous demande de choisir un Workspace (Espace de travail). Il est préférable de laisser celui par défaut. Il créer un répertoire parallèle au dossier Eclipse qui permettra d'enregistrer tous les documents des futurs projets JAVA.

Avant de continuer, on va d'abord s'assurer que le logiciel soit bien à jour !

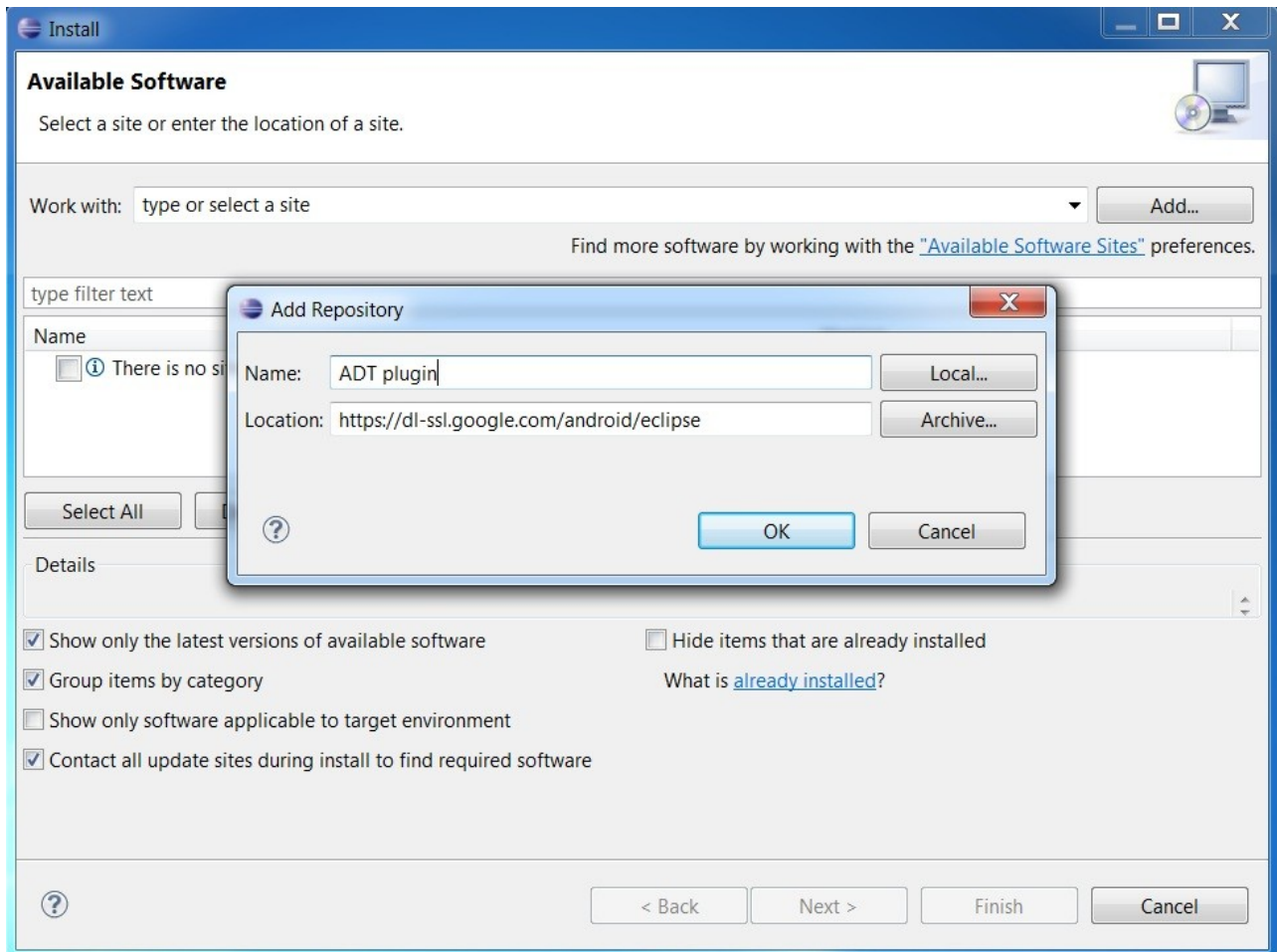
Cliquer sur le menu Help / Check for Updates

Notre Eclipse est à jour, nous pouvons continuer. La prochaine étape consiste en l'installation de l'ADT Plug-in (Plug-in ADT Android pour Eclipse).

Installation du Plugin Android pour Eclipse

Nous gardons Eclipse ouvert car les prochaines étapes se déroulent encore dans ce logiciel.
Cliquez sur le menu Help/ Install New Software...

En haut à droite de l'image vous avez un bouton Add, cliquez dessus et lorsque la fenêtre Add Repository apparaît entrez les champs suivants :



Après l'ajout de cet ADT plugin, une brève recherche s'active et donne ce résultat :

On coche « Developer Tools » puis cliquez sur « Next »

Il y a un écran Install Détails qui nous affiche tout le package de l'installation du plugin.

On peut de nouveau appuyer sur « Next »

Un écran Review licence s'affiche et il attend qu'on coche ce paramètre :

En cliquant sur « Finish » l'installation va commencer.

Un warning va s'afficher, cliquez sur « Ok »

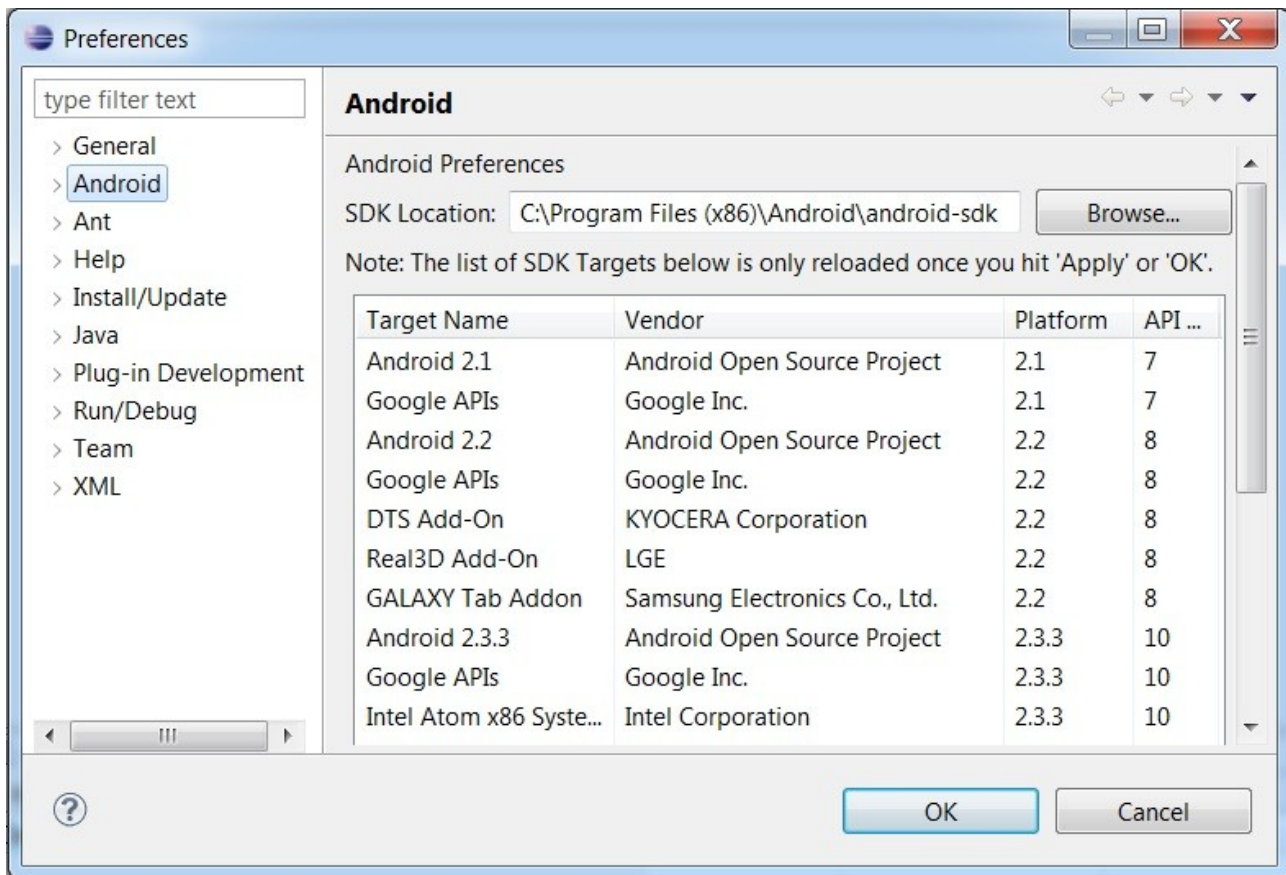
L'installation s'achève et demande le redémarrage d'Eclipse :

Cliquez sur « Restart Now »

Eclipse se redémarre, passons à l'étape suivante.

Spécifier l'emplacement du SDK Android à Eclipse

A la réouverture d'Eclipse, allez dans le menu « Window » puis « préférences »



Cliquez sur l'onglet Android comme sur l'image

Lorsque tout est bien paramétré, toutes les versions du SDK et les APIs Google installées apparaissent. Dans le cas contraire une erreur survient, cliquez sur « ok ». Cela vient d'un problème d'emplacement de fichiers : l'emplacement du **SDK Location** n'est pas trouvé.

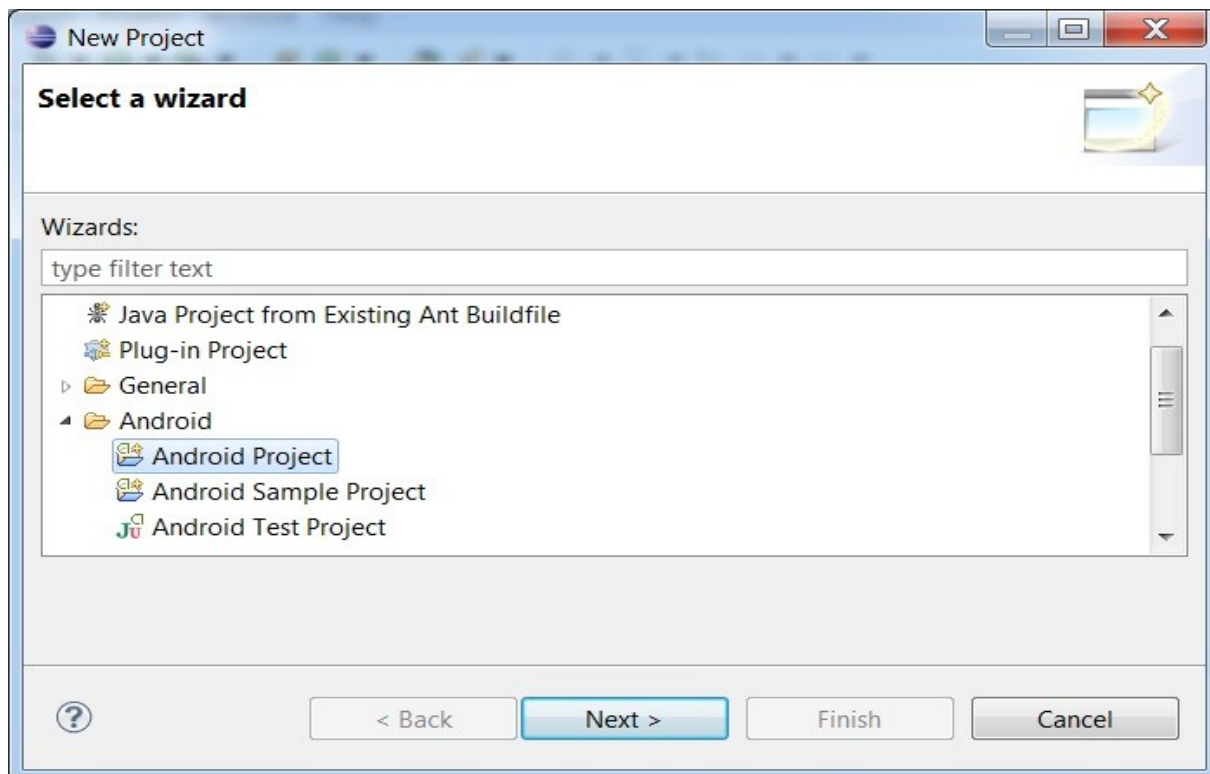
Par défaut le chemin des fichiers du SDK d'Android est correct, mais il se peut qu'Eclipse ne le trouve pas lorsque l'on installe Android SDK dans un autre dossier que celui proposé par défaut.

Il suffit tout simplement d'indiquer au logiciel le chemin en cliquant sur « Browse ».

Cliquez sur ok quand tout est bien paramétré et que toutes les versions apparaissent.

Configuration d'Eclipse pour la création d'une application

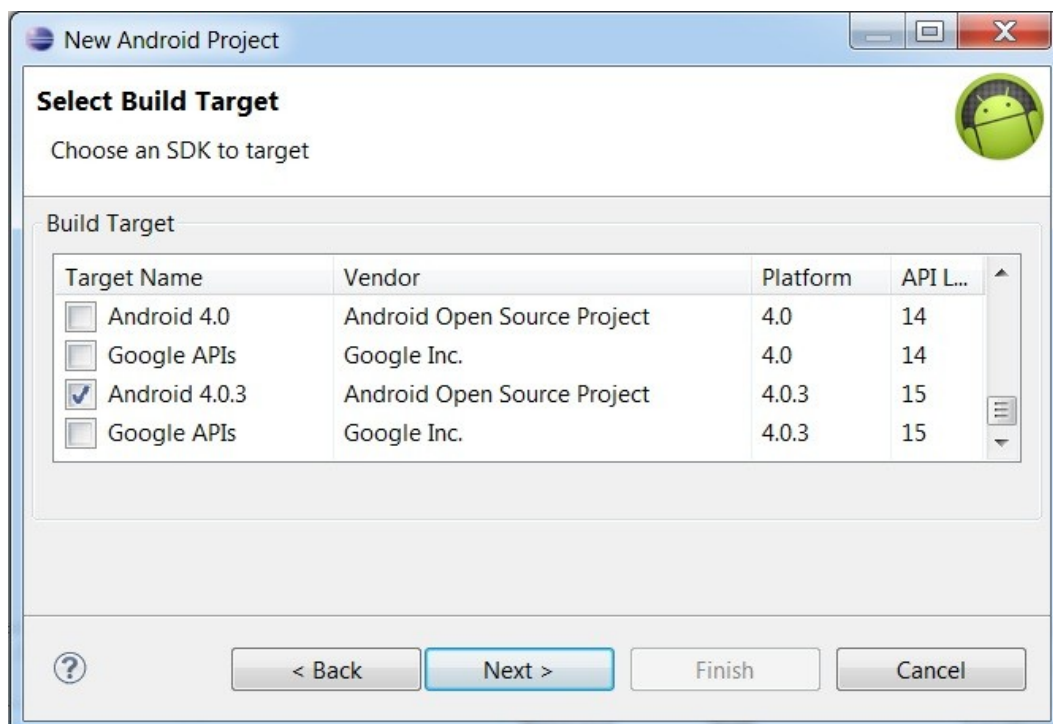
Cliquez sur le menu « **File** » puis « **new** » et enfin « **Project** » Une fenêtre s'affiche :



Sélectionnez dans le dossier Android : Android Project comme indiqué sur l'image.

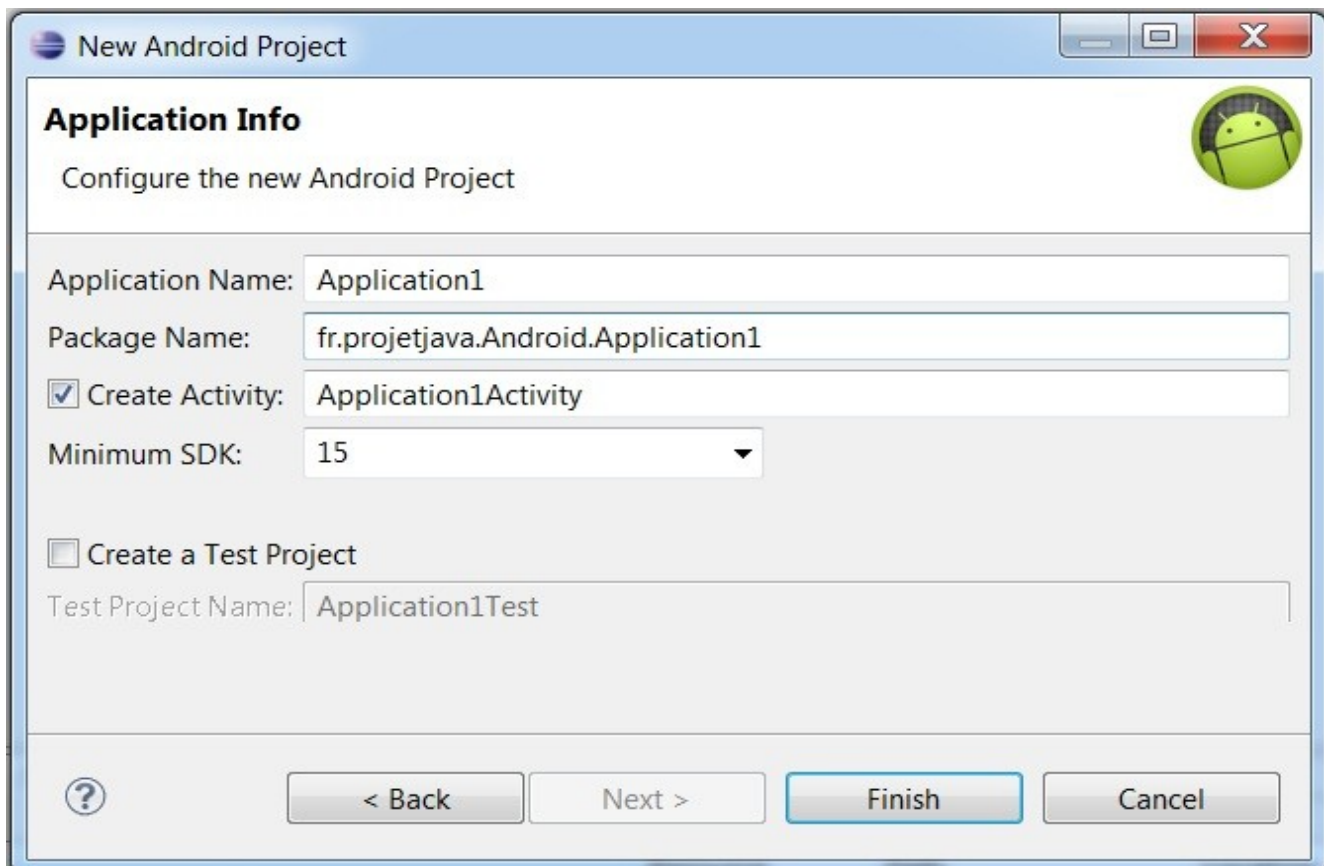
Une nouvelle fenêtre s'affiche : Remplir le champ « **Project Name** ».

Ne pas changer les autres champs.



Une nouvelle fenêtre s'affiche :

Par défaut Eclipse coche au préalable la dernière version en date. Mais on est libre de choisir les différentes versions selon nos souhaits. On clique sur « Next »



Définition du Package Name : Nom du package principal de l'application. Le nom du package suit, en général, la règle suivante :

extension_de_votre_domaine.nom_du_domaine.android.nom_du_projet.

Exemple : fr.projetjava.Android.Application1

Le package définira en fait le nom interne de l'application sur le terminal Android.

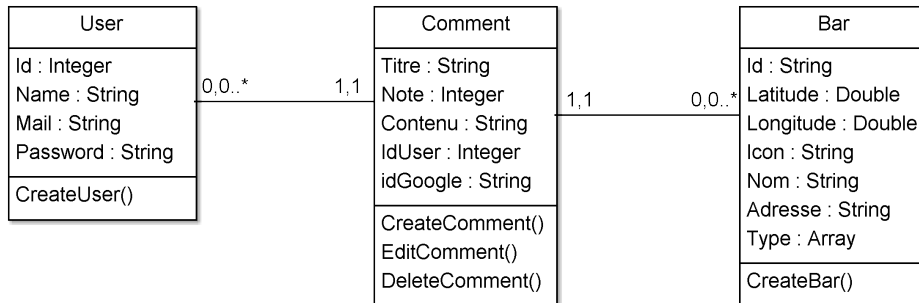
Une fois le tout bien paramétré, il ne reste plus qu'à cliquer sur « Finish »

Vous êtes désormais prêt à développer sur Android.

II. Conception du projet

1. Schématisation UML

Diagramme de classes



L'utilisateur peut déposer un ou plusieurs commentaires ;

Un commentaire est déposé par un et un seul utilisateur et concerne un et un seul bar ;

Un bar peut avoir plusieurs commentaires.

2. Présentation des tables

Table utilisateur :

Champs :

ID : Identifiant de la table utilisateur.

NAME : Nom de l'utilisateur.

FIRSTNAME : Prénom de l'utilisateur.

MAIL : Mail de l'utilisateur.

PSEUDO : Pseudo de l'utilisateur.

Fonctions :

Insertion : Permet l'inscription des différents utilisateurs de l'application.

Table commentaire :

Champs :

ID : Identifiant de la table commentaire

TITRE : Titre du commentaire

NOTE : Note de l'utilisateur sur l'endroit concerné

CONTENU : Contenu du commentaire

IDUSER : Clé étrangère vers la table utilisateur

IDGOOGLE : Identifiant google maps du lieu concerné.

Fonctions :

Insertion : Permet la création d'un commentaire sur un lieu

Edition : Permet la modification d'un commentaire.

Suppression : Permet la suppression d'un commentaire.

2. Technologies utilisées

Eclipse :

Eclipse est un IDE, mais surtout une plateforme pour héberger les applications. Les IDE Java open source sont rares. La caractéristique essentielle d'Eclipse est l'extensibilité de l'environnement.

Plus que de se focaliser sur un environnement de développement Java, les concepteurs d'Eclipse se sont efforcés avant tout de créer un socle applicatif.

Le coeur d'Eclipse est en fait composé :

1. D'un socle capable de charger des modules (plugins)
2. De modules de base permettant de gérer des ensembles de ressources (projets, fichiers, répertoires, ...)
3. De modules pour permettre la création d'interfaces graphiques cohérentes. En utilisant cette librairie, les plugins ont un aspect homogène.
4. La partie développement Java n'est qu'un ensemble de plugins qui sont la première utilisation de ce socle.

MySQL :

Le logiciel MYSQL est un serveur de base de données (endroit où l'on range plein d'informations) SQL très rapide, Multi-Threadé, Multi-Utilisateur et robuste.

Le serveur MYSQL est destiné aux missions stratégiques et aux systèmes de production à forte charge, ainsi qu'à l'intégration dans des logiciels déployés à grande échelle.

MYSQL est à ce jour suffisamment élaboré pour être utilisé dans de très nombreux cas. Sa fiabilité est généralement excellente, sa disponibilité totale, et en plus il est gratuit. MYSQL permet de créer des tables contenant plusieurs milliards d'enregistrements.

SDK Android :

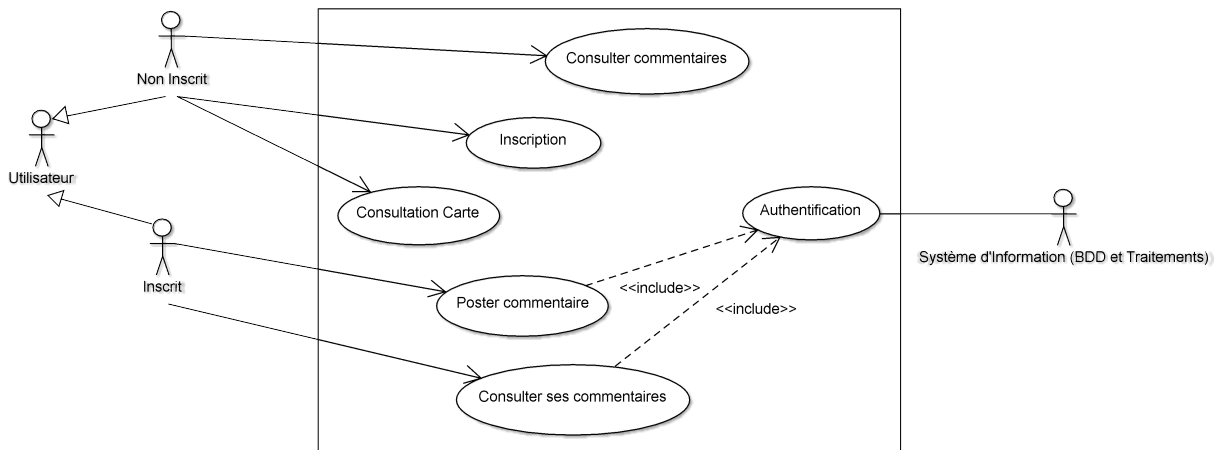
Android est un système d'exploitation open source³ utilisant le noyau Linux, pour smartphones, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs et des tablettes.

III. Analyse technique

1. Approche globale

L'application repère la position GPS de l'utilisateur et lui permet d'obtenir les adresses des endroits dans un rayon de 500m d'où il se trouve. L'utilisateur peut par la suite déposer des commentaires.

Diagramme de cas d'utilisation



2. Problématique BDD

La gestion des bases de données avec Android n'est pas intégrée.

Il faut donc utiliser des scripts PHP, ces scripts envoient la requête à la base de données et renvoie les données sous forme de json. Json qui doit être interprété par notre application.



json



HTTP



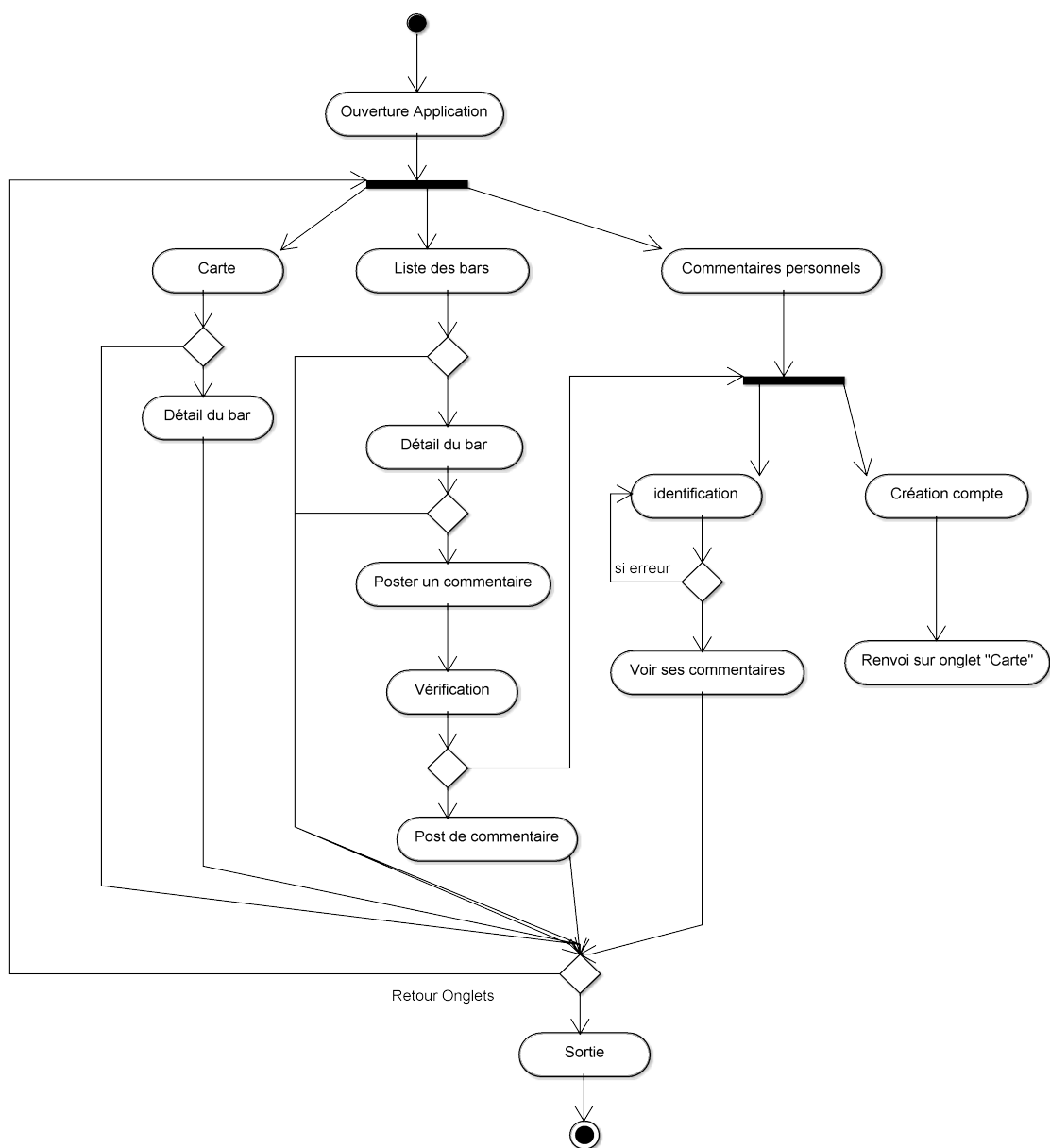
3. Problématique Outil Google

Nous utilisons Google Maps pour ce qui est de la localisation. En passant différents paramètres de données de localisation GPS de l'utilisateur, Google Maps permet de retourner une liste de tous les lieux correspondant à la recherche. Donc les bars présents autour de l'appareil Android.

Il faut tout d'abord obtenir une clé API auprès de Google.
Cette clé API est nécessaire pour le bon fonctionnement d'un service comme Google Maps.

IV. Mise en œuvre

1. Diagramme de cas d'utilisation



2. Implémentation, description des différentes classes

MainActivity.java

Une Activity est une vue pour Android.

Cette classe est la classe maitresse de l'application. C'est en quelque sorte l'accueil de l'application. Dans cette classe les onglets sont créés. Elle hérite de TabActivity, classe nécessaire à la création de tableaux donc de nos onglets.

Les onglets sont instanciés en tant qu'objets TabSpec.

Nous avons donc trois onglets :

tabSpectMap qui permet d'afficher la carte google maps ;

tabSpectComment qui permet de poster un commentaire sur une adresse particulière ;

tabSpectCommentFromUser qui permet de voir tous les commentaires postés par l'utilisateur.

Login.java

Cette classe gère la connexion de l'utilisateur.

Elle hérite d'Activity et permet la gestion des connexions.

Soit l'utilisateur dispose déjà d'un compte, dans ce cas là, il peut se connecter à l'application.

Soit il ne dispose pas de compte et peut en créer un directement.

La vérification de login / password passe également par un script PHP, Login.php

MapTutorialActivity.java

Cette classe gère l'affichage de la carte google maps.

On fait appel aux différentes méthodes nécessaires au bon fonctionnement de cette vue.

CommentFromUser.java

Cette classe gère l'affichage de tous les commentaires de l'utilisateur.

On récupère l'id de l'utilisateur, avec cette variable, nous pouvons retrouver les différents commentaires de l'utilisateur à l'aide du script PHP suivant.

getCommentFromIDUser.php

L'interface entre la base de données et l'application Android se fait généralement via PHP.

Ce script PHP permet de retourner tous les commentaires en base de données de l'utilisateur sélectionné.

Il retourne en JSON le résultat de la requête. Résultat qui est traité au niveau de l'application Android.

BarsController.java

Cette classe gère la transmission des coordonnées nécessaires à la recherche google maps.

On passe en paramètre la latitude et la longitude.

Conclusion :

La réalisation de ce projet nous a permis de gérer la conception d'un projet de A à Z.

En effet, nous avons pu mettre en œuvre tous les cours que le CNAM nous a dispensé en matière de modélisation UML par exemple, mais aussi les cours de Base de Données.

L'utilisation d'Android nous a permis d'utiliser certains concepts Java vus au cours de l'année et d'utiliser Eclipse, IDE complet qui permet le développement dans de nombreux langages de programmation.

Android reste assez élitiste au niveau de la programmation. Cela nécessite de nombreuses heures de pratique et de lecture de documentation afin d'être opérationnel. La communauté est très active ce qui permet de régler de nombreux problèmes et de compenser la gestion d'erreurs un peu chaotique.

Nous avons également utilisé l'API Google MAPS. L'utilisation d'une API Google reste intéressant d'un point de vue professionnel et personnel. En effet, de plus en plus d'applications utilisent ces technologies de localisation et les technologies Google en général (Google Agenda, Gmail, etc...). Il reste intéressant de comprendre le fonctionnement d'une telle application.