

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

***Création d'une application android via
Une autre application android (partie serveur)***

Réalisé par :

- SENOUSSAOUI Ikram.
- LAKEHAL Asma.

Présenté le Juin 2013 devant la commission d'examination composée de MM.

- SMAHI MED Ismail (Encadreur)
- BENTAALLAH Mohamed (Examineur)
- ABDELLAOUI Ghoutti (Examineur)

Année universitaire: 2014-2015.

Remerciement

C'est avec un grand plaisir qu'on réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail.

Tout d'abord on remercie notre encadreur Monsieur SMAHI ISMAIL pour son soutien, son sérieux, sa disponibilité, ses précieux conseils et son aide tout au long de l'élaboration de ce travail.

Nous remercions les membres de jury qui ont bien voulu examiner et évalué ce mémoire.

Nous nous acquittons, enfin, volontiers d'un devoir de gratitude et de remerciements à tous nos enseignants pour la qualité de l'enseignement qu'ils ont bien voulu nous prodiguer durant nos études afin de nous fournir une formation efficiente.

Dédicaces

A nos chers parents,

Que nulle dédicace ne puisse exprimer ce que nous leurs doit, pour leur bienveillance, leur affection et leur soutien... Trésors de bonté, de générosité et de tendresse, en témoignage de nos profonds amours et nos grandes reconnaissances « Que Dieu vous garde ».

A nos chères sœurs et nos frères,

En témoignage de nos sincères reconnaissances pour les efforts qu'ils ont consenti pour l'accomplissement de ce projet. On leur dédie ce modeste travail en témoignage de notre grand amour et notre gratitude infinie.

A tous nos amis,

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

A toute nos Familles

A tous ceux dont l'oubli du nom n'est guère celui du cœur...

« De l'union « si » avec « mais » naquit enfant nommé « jamais » »

« Il n'y a pas de « si » ni de « mais », il faut réussir »

Table de matière

Table des matières

Remerciement.....	2
Dédicaces	2
Table de matière.....	4

CHAPITRE 1 (introduction à l'android)

I. Introduction.....	7
I.1. L'histoire de l'android.....	7
I.2. C'est quoi l'android.....	8
Conclusion chapitre1 :.....	9

CHAPITRE 2 (Ingénierie Dirigée par Modèles):

I. Introduction	10
I.1. Définition	10
I.2. C'est quoi le model	10
I.3. Qu'est-ce que la modélisation ?.....	11
I.4. Les avantages de la modélisation :	11
I.5. les outils de L'IDM	11
I.6 : Diagramme de cas d'utilisation.....	14
I.7 : Récapitulant le modèle et le méta modèle	15
Conclusion chapitre2.....	15

CHAPITRE 3 (la mise en œuvre)

III-1. La partie serveur	16
III-1.1 La réception et la récupération des informations	16
III-2.2. La création du projet android :.....	17
III-1.3 La modification du projet :.....	20
III-1.3.1 Modification de la classe MainActivity .java:.....	20
III-1.3.2 Modification de la classe activitymain.xml:	20

Table de matière

<i>III-1.3.3 Modification de la classe AndroidManifest.xml:</i>	21
<i>III-1.3.4 Modification du String.xml: ations</i>	21
<i>III-1.3.5 Ajouter le répertoire drawable :</i>	22
<i>III-1.3.6 Ajouter le répertoire Menu:</i>	22
<i>III-1.4 La compilation du projet :</i>	23
<i>III-1.5 Envoyer l'APK:</i>	25
<i>III-1.6 La classe main du serveur :</i>	31
Conclusion générale.....	32
Perspectives	33

Introduction générale

Des milliers (voir beaucoup plus !) de personnes (plus ou moins professionnelles) développent des applications sous Android.

Beaucoup de ces applications se retrouvent sur le grand supermarché des applications Android mis en place par Google : l'Android Market.

Des applications sont gratuites, d'autres, payantes (en général, quelques euros), certaines, sont de grande qualité, d'autres, beaucoup moins.

Il est relativement "facile" de créer une application (simple) pour Android (ceci explique sans doute le nombre d'applications de qualité très "moyenne" que l'on trouve sur l'Android Market !), il "suffit" d'avoir de bonnes bases en programmation Java , et d'être prêt à faire les efforts nécessaires pour acquérir certaines méthodes particulières au développement sous Android.

Le présent rapport est organisé comme suit :

Nous allons voir dans le premier chapitre une introduction pour android, par la suite, l'Ingénierie dirigé par des modèles dans le chapitre 2.

Le chapitre3 est réservé pour détailler les différentes étapes de réalisation de la partie serveur.

Chapitre1 : Introduction à l'android

I. Introduction

Ce chapitre est une introduction à la plate-forme ANDROID¹, on va parler un petit peu sur l'histoire de cette plate-forme, ces versions, ces bibliothèques en répondant aussi la question « Android, c'est quoi, ça veut dire quoi, ça fait quoi ? Et Qu'est ce que c'est ? »

I.1. L'histoire de l'android

La conception d'un OS² mobile par Android été en octobre 2003 à Palo Alto par Andy Rubin (une société américaine nommé ANDROID), et en Aout 2005, GOOGLE a racheté cette dernière. L'objectif premier était de développer un système d'exploitation qui permettrait à l'utilisateur d'interagir avec les OS, dans le temps ou tous les systèmes sont embarqués

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en septembre 2008

Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités, le tableau suivant résume quelques versions de l'android :

<u>La version :</u>	<u>La date d'apparition :</u>
Android 1.0	Septembre 2008
Android 1.1	Février 2009
Android 1.5 dit « Cupcake »	Avril 2009
Android 1.6 dit « Donut »	Septembre 2009
Android 2.0 dit « Eclair »	Octobre 2009
Android 2.0.1	Décembre 2009
Android 2.1, 2.2, 2.3	Janvier - Décembre 2010
Android 2.3.3, 2.3.4, 2.3.5, 2.3.6,2.3.7	Février-septembre 2011
Android 3.0	Février 2011
Android 3.1	Mai 2011
Android 3.2	JUILLET 2011
Android 3.2.1, 3.2.2	Septembre 2011
Android 4.0	Octobre2011
Android 5.0	Octobre2014
Android 5.1	Mars 2015

¹Système d'exploitation.

² Operating System (un système opérateur).

1.2. C'est quoi l'android

Beaucoup de fausses vérités circulent sur le web à propos d'Android. Android c'est une plateforme complète pour appareil mobile, lancé par Google et l'OHA³ qui compose d'opérateurs mobiles, de fabricants de téléphones et d'éditeurs logiciels. Il concurrence des plateformes telles que l'iOS⁴ d'Apple⁵, Windows Mobile de Microsoft, RIM⁶ intégré dans les BlackBerry, Symbian⁷ et MeeGo⁸ de Nokia.

La plateforme Android est un OS (Operating System) basée sur un kernel⁹ linux entièrement gratuit, sous licence open source. Elle est composée d'un système d'exploitation, de bibliothèques, et d'un ensemble d'applications

Rappelons les points clés d'Android en tant que plate-forme :

- elle est innovante car toutes les dernières technologies de téléphonie y sont Intégrées : écran tactile, accéléromètre, GPS¹⁰, appareil photo numérique etc.
- elle est accessible car en tant que développeur vous n'avez pas à acheter de matériel spécifique (si vous voulez aller plus loin que l'utilisation d'un émulateur, un téléphone Android pour effectuer vos tests vous sera toutefois nécessaire), ni à connaître un langage peu utilisé ou spécifique : le développement sur la plateforme Android est en effet réalisé en langage Java, un des langages de programmation les plus répandus
- elle est ouverte parce que la plate-forme Android est fournie sous licence open source, permettant à tous les développeurs – et constructeurs – de consulter les sources et d'effectuer les modifications qu'ils souhaitent.

Les composants majeurs de la plate-forme Android sont résumés sur le schéma suivant (Traduit de la documentation Google) :

³ Open Hanset Alliance :un consortium de plusieurs entreprises dont le but est de développer des normes ouvertes pour les appareils de téléphonie mobile.

⁴ originally iPhone OS

⁵ conçoit et réalise l'iPhone, l'iPad, les ordinateurs portables et de bureau Mac, iOS 8, OS X, l'iPod et iTunes..

⁶ Research In Motion

⁷ Symbian OS est un système d'exploitation pour téléphones portables.

⁸ un système d'exploitation et une plate-forme de développement pour les appareils mobiles. Celui-ci est opensource et basé sur Linux.

⁹ Un noyau de système d'exploitation

¹⁰ Le Global Positioning System (GPS) est un système de géolocalisation fonctionnant au niveau mondial.

Chapitre1 : Introduction à l'Android

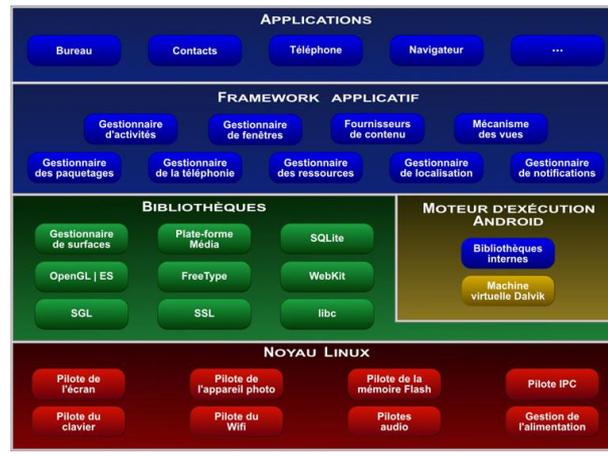


Figure 1 : Les composants d'android (m)

Conclusion

Dans le chapitre 1, nous avons parlé un petit peu sur le system d'exploitation android, maintenant, nous allons passer au 2^{ème} chapitre pour voir l'approche IDM.

Chapitre2 : Ingénierie Dirigée par Modèles

I. Introduction

Dans ce chapitre on va voir un domaine d'informatique qui est très utilisé dans la programmation c'est : **IDM¹¹ ou MDE¹²**, alors c'est quoi cet ingénierie et pour quoi on l'utilise ?

I.1. Définition

L'IDM (Ingénierie dirigée par les modèles) est le domaine d'informatique *mettant à disposition des outils, concepts et langages pour créer et transformer* des modèles. Ce que propose l'approche d'ingénierie des modèles (IDM, ou MDE en anglais pour Model Driven Engineering) est simplement de mécaniser le processus que les ingénieurs expérimentés suivent à la main, et de modéliser les applications à un haut niveau d'abstraction.

Ses Langages sont : UML¹³, MOF¹⁴, QVT¹⁵, ECLIPSE.

I.2.C'est quoi le modèle

Un modèle est une abstraction d'objets de la réalité. C'est donc une simplification du monde réel. La problématique consiste alors à trouver le bon niveau d'abstraction et à exprimer les concepts du monde réel dans un langage assez abstrait mais aussi précis qu'un langage de programmation, pour que ce langage soit interprétable par un programme informatique.

En d'autre terme : «En informatique, un modèle a pour objectif de structurer les données, les traitements, et les flux d'informations entre entités.»

¹¹ Ingénierie Dirigée par les Modèles

¹² Model Driven Engineering

¹³ *Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML).*

¹⁴ *Meta-Object Facility en Génie logiciel .*

¹⁵ *Query/View/Transformation est un standard défini par l'OMG.*

I.3.Qu'est-ce que la modélisation ?

Modéliser consiste à décrire dans un langage graphique formel, des besoins, les solutions fonctionnelles et techniques qui leur correspondent afin de bénéficier des atouts évidents qu'apporte un bon modèle, il est nécessaire que les personnes qui les réalisent et que celles qui les utilisent soient convenablement formées et accompagnées et travaillent selon le même cadre méthodologique.

I.4.Les avantages de la modélisation :

Donc ceci nous ramène à posé la question suivante quelle sont les avantages de la modélisation ??

Un bon modèle permet :

- Lors du développement : de simplifier la communication entre les intervenants d'un projet
- Lors de la maintenance : de bénéficier d'une source d'information garantissant une analyse d'impact rapide et une meilleure qualité de réalisation.
- Abstrait : Il fait ressortir les points importants tout en enlevant les détails non nécessaires
- Compréhensible : Il permet d'exprimer une chose complexe dans une forme plus facilement compréhensible par l'observateur
- Précis : Il représente fidèlement le système modélisé
- Prédictif : Il permet de faire des prévisions correcte sur le système modélisé

I.5.les outils de L'IDM

On trouve des différentes outils utilisé dans L'ingénierie diriger par model on va citer deux :

I.5.1 Eclipse :

Eclipse est un EDI¹⁶ , ou en anglais un IDE¹⁷ .

Développé par IBM à partir de ses ancêtres Visual Age et Visual Age For Java, il a depuis été rendu open-source et son évolution est maintenant gérée par la Fondation.

Sa conception est complètement modulaire : basée sur des notions telles que "micro noyau OSGi¹⁸" (depuis la version 3) ou encore les plug-ins, ce qui fait d'Eclipse une boîte à outils facilement modifiable et extensible.

¹⁶ Environnement de Développement Intégré.

¹⁷ Integrated Development Environment

Eclipse est également devenu une plate-forme(Framework pour manipuler modèle et méta modèle),servant de socle à d'autres applications, que celles-ci soient destinées au développement logiciel ou à tout autre domaine. Cette plate-forme est nommée Eclipse RCP¹⁹ .

Eclipse a pour objectif de produire et de fournir des outils pour la réalisation de logiciels, il été conçu uniquement pour produire des environnements de développement, cependant les utilisateurs et les programmeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes.

Alors, on utilise cette Platform pour développer les applications android, et ça ce qu'on a fais dans notre projet on intégrant ce qu'on appelle le SDK

Eclipse et android :

Les applications Android sont développées en Java, mais un appareil sous Android ne comprend pas le Java tel quel, il comprend une variante du Java adaptée pour Android. Un SDK, un **kit de développement** dans notre langue, est un ensemble d'outils permettant de développer pour une cible particulière. Par exemple pour développer pour une console de jeu vidéo, on utilise un SDK spécifique pour développer des applications pour cette console. Le SDK Android est donc un ensemble d'outils que met à disposition Google afin de vous permettre de développer des applications pour Android.

Les outil SDK sont nécessaire pour pouvoir utilisé éclipse.

1.5.2Language UML :

UML est le standard pour la modélisation des systèmes informatiques, 12 années d'utilisation d'UML, ont permis aux experts Viseo d'en identifier les bonnes pratiques : celles qui permettent de tirer le meilleur de la notation UML, sans les pièges que peuvent être la lourdeur, la complexité, ou la désynchronisation avec la réalité du logiciel en cours de développement.

Les activités d'un projet informatique :

Quelle que soit la méthode employée, on retrouve toujours les activités ci-dessous dans un projet informatique. Les différences résident dans leur degré de formalisation et leur

¹⁸ Open Services Gateway initiative est une organisation qui spécifie une plate-forme de services fondée sur le langage Java qui peut être gérée de manière distante.

¹⁹ Rich Client Platform

orchestration : itérative, séquentielle, incrémentale, pilotée par les modèles, pilotée par les tests, etc...

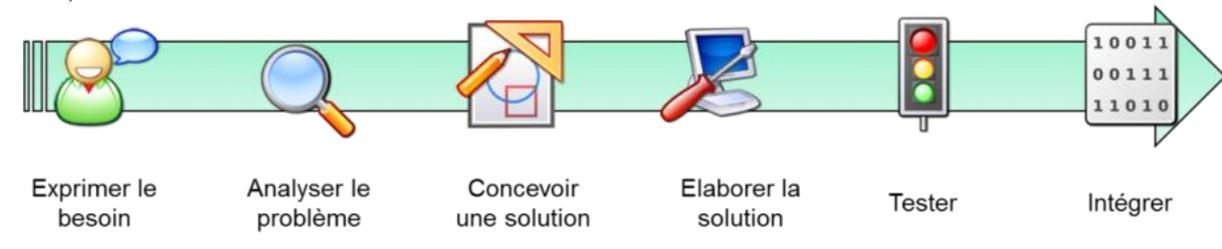


Figure 2: Les étapes de conception du projet

Expression du besoin :

C'est l'activité qui consiste à décrire de façon précise le logiciel qui correspond aux besoins des utilisateurs et des prescripteurs. Ceux-ci peuvent être formalisés dans un modèle qui permettra par la suite de gérer une traçabilité avec les modèles de conception, permettant de faire rapidement des analyses d'impact.

Analyse :

C'est l'activité qui consiste à comprendre précisément le besoin décrit lors de son expression et à le compléter en utilisant différentes sources d'information. Elle résulte dans la création d'un modèle d'analyse qui est le point d'entrée de la conception.

Conception :

C'est l'activité qui consiste à définir une solution aux besoins synthétisés dans le modèle d'analyse. Elle consiste principalement à bien répartir les responsabilités entre les différents composants – classes, composants, sous-systèmes - en respectant les principes fondamentaux de toute bonne conception : un faible couplage et une forte cohésion.

Les diagrammes de package pour structurer l'application, les diagrammes de classe pour définir les API des composants ou détailler une structure particulièrement riche d'un composant sont les diagrammes les plus utilisés.

Cette démarche, fréquemment basée sur le standard UML comme notation de référence sur le projet, fait ses preuves dans de nombreuses entreprises auxquelles elle permet :

- De structurer le processus de développement logiciel pour une meilleure productivité
- D'augmenter la qualité des logiciels développés

- D'assurer une bonne traçabilité entre l'expression des besoins et les artefacts constituant le logiciel final (sources, procédures d'installation, documentation, etc.)
- D'assurer une maintenabilité du logiciel qui s'appuie sur un référentiel simple, précis et à jour, et pas seulement sur la mémoire des développeurs ou sur le code lui-même : « code never lies », certes, mais il est très souvent composé de plusieurs dizaines ou centaines de milliers de lignes, et donc d'un niveau de détail incompatible avec les besoins d'évolutions et d'études d'impacts que tout logiciel nécessite.
- Enfin de faciliter la stratégie de test automatisé, pratique indispensable aux méthodes agiles couplées à l'intégration continue.

1.6 :Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les services les plus importants rendus par un système.

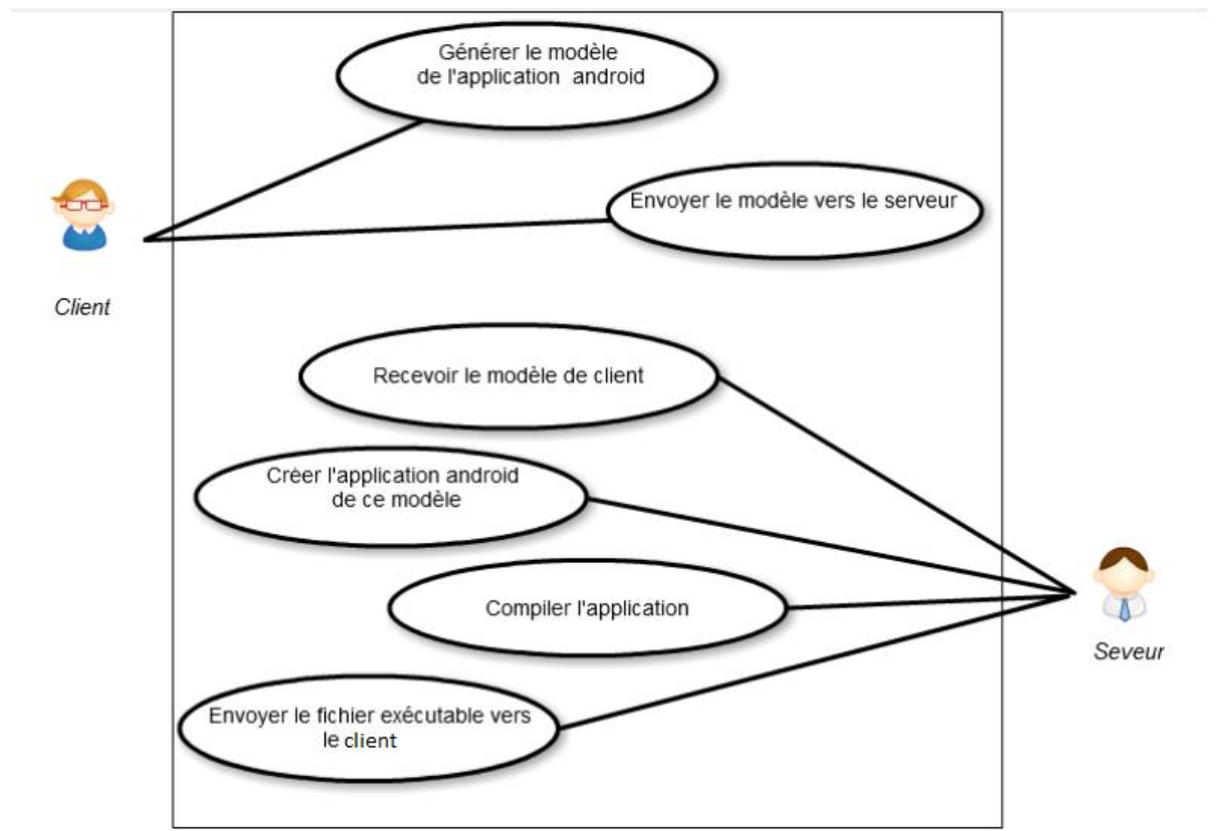


Figure 3:Le diagramme de cas d'utilisation

1.7 : Récapitulant le modèle et le méta modèle

- Méta-méta modèle :
 - Langage pour décrire des langages.
 - Ce décrit lui même.
- Méta modèle :
 - Langage pour décrire des modèles.
- Modèle :
 - est une représentation de la réalité, il a pour objectif de structurer les données, les traitements, et les flux d'informations entre entités.
- Monde réel.

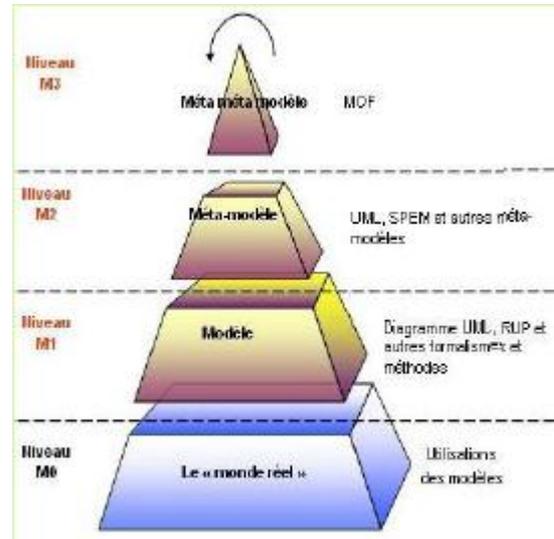


Figure 4:diagramme IDM (I)

Conclusion

Dans ce chapitre, nous avons vu le concept de l'ingénierie dirigée par des modèles et ses objectifs, passant maintenant au chapitre suivant pour expliquer les étapes de la création de l'application client.

Chapitre3 : La mise on œuvre

III-1.La partie serveur

Le serveur, c'est la partie qui représente l'application java, Le but ici c'est de créer l'application android suivant les données envoyées par le client, alors on crée un projet android, le modifie, le compile pour générer le Apk et ensuite envoyer ce dernier par une socket vert le client approprier.

III-1.1 La réception et la récupération des informations

III-1.1.1 La réception des informations (h)

La première étape pour construire un nouveau projet pour un client c'est d'abord la réception du fichier XML²⁰ « Information.xml » qui contient toutes les informations déjà entré par l'utilisateur

Donc pour récupérer le fichier on a implémenté la classe « ReceptionFichierXml » qui utilise la méthode lancer () pour le traitement.

- on va mettre en œuvre une sorte de prise que les serveurs peuvent utiliser pour écouter et accepter les connexions des clients, c'est la socket Ce qui nous donne :

```
ServerSocket serverSocket = new ServerSocket(port);
```

- Après tentative de connexion, si tout va bien, le serveur accepte la connexion du client, et reçoit un nouveau socket qui est directement lié au même port local. Il a besoin d'une nouvelle prise de sorte qu'elle puisse continuer à écouter le socket d'origine pour les demandes de connexion, tout t'en satisfaisant les besoins du client connecté. Voici comment accepter la connexion d'un client :

```
socket = serverSocket.accept();
```

²⁰ Extensible Markup Language.

- Une fois la connexion établie et **les sockets en possession**, il est possible de récupérer le flux d'entrée et de sortie de la connexion TCP²¹ vers le serveur.

```
InputStream is = socket.getInputStream();
```

Et finalement, Enregistrer le fichier reçu comme « OutputStream » pour l'utiliser dans la création du nouveau projet.

III-1.1.2 La récupération des informations :

On a le fichier qui contient les informations, il nous faut maintenant les récupérer, pour cela on implémente la classe « RecupererLesInformationsClient » qui utilise « red » comme méthode.

Dans cette classe :

- on accède au fichier « information.xml » et on l'analyse.
- l'analyse du fichier s'effectue par le « parser » et la normalisation du fichier. (a)
- et on récupère les éléments pour les utiliser dans la partie suivante.

III-1.2.La création du projet android :

Générer des applications pour d'autres plateformes, nécessite l'installation du **SDK**²² pour chaque plateforme visée.

Pour cela avant de commencer la création du projet android, on met en place la variable **ANDROID_HOME**, attention c'est bien le **SDK** qu'il faut référencer et pas le répertoire contenant Eclipse et le **SDK**.

Démarrer >panneau de configuration >Système >Paramètres système avancés > Variable d'environnement

Et on crée une nouvelle variable :

ANDROID_HOME="c:\path\to\android\jdk"

²¹ Transmission Control Protocol

²² Software development kit

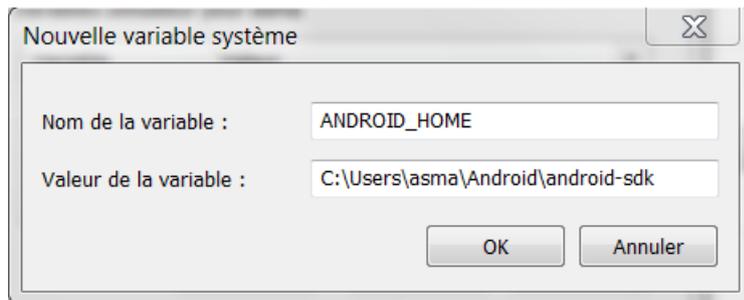


Figure 5 : Modification de la variable ANDROID HOME

Et on l'ajoute dans le « Path » du system .

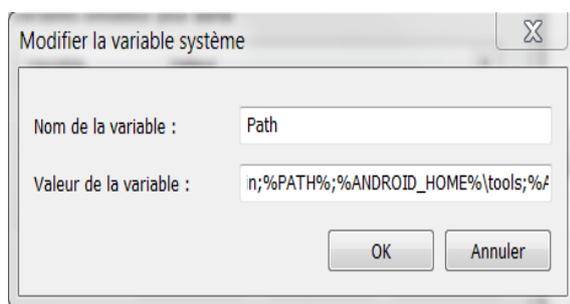


Figure 6:Modification de la variable PATH

Maintenant on passe la création du notre projet ,on utilise pour cela un fichier batch (*.bat) qui contient les commandes DOS suivantes :

```
android create project --target 1 --path nom_de_l'application --activity  
MainActivity --package com.exemple.nom_de_l'application
```

(f)

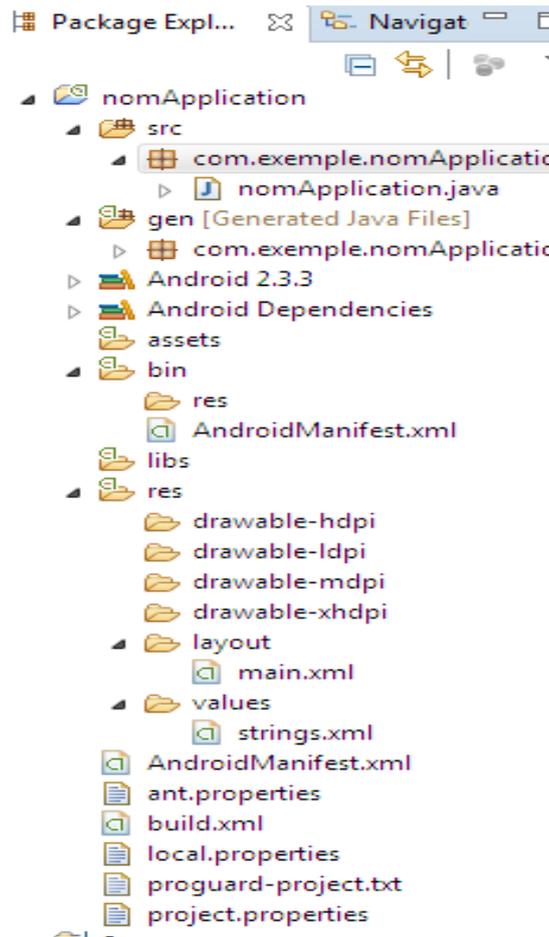


Figure 7:Arborescence d'un nouveau projet Android

Nous ne rentrerons pas ici dans les détails de cette arborescence, mais voici tout de même dans les grandes lignes les choses intéressantes :

- Les fichiers source Java sont situés dans le répertoire “src”.
- Tout ce qui concerne les chaînes de caractères, fichiers XML représentant l’interface graphique (ou “Layouts”) ou images sont dans le répertoire “res”.
- Le dossier «gen» les fichiers générés automatiquement lors de la compilation.
- Le fichier « AndroidManifest.xml » est chargé de décrire l’application au système Android. On y trouve le nom du package Java, les versions minimum et maximum du SE avec lesquelles l’application est compatible, les déclarations des permissions pour autoriser l’application à accéder à Internet par exemple, ou encore les différentes composantes de l’application qui vont être utilisées.

III-1.3 La modification du projet :

La modification consiste à attribuer quelques changements dans le projet crée en utilisant des commandes.

Commençant par cité ses modifications :

III-1.3.1 Modification de la classe MainActivity.java:

Dans cette partie, on a accéder à la classe MainActivity.java du nouveau projet en spécifiant le chemin approprié, afin de vider sans contenu avec la méthode « main » de la classe « Vider » et ensuite, d'ajouter des instructions à la classe c a d :

- L'ajout des boutons qui représente les services choisis par le client, ainsi les « setOnClick » pour chaque bouton, et aussi tous les imports et les méthodes qui caractérisent la classe MainActivity.java.
Ces modifications sans regroupés dans une classe « ModifyMainjava » dans notre serveur. (c)

III-1.3.2 Modification de la classe activitymain.xml:

On va accéder dans cette étape aussi à l'activitymain.xml du nouveau projet, pour vider son contenu en utilisant la méthode « XML » de la classe « Vider », et la remplir en instanciant les composants définis dans la classe java, et le plus important aussi c'est de définir l'interface de la nouvelle application.

- Ajouter les textes view, une image pour l'interface de l'application, les boutons qui représente les services demandés, et le bouton qui permet de se connecté à notre site web
- Pour faire ses modifications on est besoin du « DOM » (httdom) pour créer la structure du fichier XML, voici un exemple de cette classe :

```
dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.newDocument();

Element rootElement = doc.createElementNS(uri, "LinearLayout");
rootElement.setAttribute("xmlns:android", uri);
rootElement.setAttribute("android:layout_width", "fill_parent");
rootElement.setAttribute("android:layout_height", "fill_parent");
rootElement.setAttribute("android:orientation", "vertical");
rootElement.setAttribute("android:background", "@drawable/image");

doc.appendChild(rootElement);
```

Ces modifications sans regroupés dans une classe « ModifyMainXML» dans notre serveur.

III-1.3.3 Modification de la classe AndroidManifest.xml:

On a attribué aussi une modification sur le fichier manifest.xml du nouveau projet ,en ajoutant un attribue qui définit l'api de l'application en utilisant la classe « ModifyManifest ».

- Accéder au fichier Manifest.xml du nouveau projet et l'analyser on utilisant l'instruction :

```
Document doc = dBuilder.parse(xmlFile);  
doc.getDocumentElement().normalize();
```

- Appeler la méthode « ModifyAttribue » de la classe « ModifyManifest » pour ajouter l'attribue.
- Et finalement diriger le résultat dans le fichier « AndroidManifest.xml » de la nouvelle application.

III-1.3.4 Modification du String.xml: ations

Dans cette classe on va modifier la valeur de l'attribue « name » pour qu'elle prenne le nom de la nouvelle application, et ajouter l'élément suivant :

```
<string name="action_settings">Settings</string>
```

- Accéder au fichier Sring.xml suivant le chemin défini dans le programme
- Analyser le fichier et appeler les méthodes « update » et « AddElement »
Qui sert successivement à modifier l'attribue et ajouter l'élément qu'on a déjà parler.
- Et finalement diriger le résultat dans le fichier « String.xml » de la nouvelle application.

Ainsi on a ajouter des répertoires qui manque pour pouvoir compiler notre projet

III-1.3.5 Ajouter le répertoire drawable :

Pour ajouter ce répertoire on a implémenté la classe « AddDrawable » qui utilise la méthode « create ».

- Cette méthode se charge à créer le répertoire « Drawable», et elle fait appelle aussi à la méthode « CopyFile », elle prend le chemin ou on veut créer le répertoire comme paramètre.
- La méthode « CopyFile » , c'est la méthode qui sert copier l'image de l'interface dans le répertoire crée .

III-1.3.6 Ajouter le répertoire Menu:

La classe « AddMenu » est responsable de l'ajout du répertoire Menu dans le nouveau projet.

- Cette classe utilise « done » comme méthode, ainsi on trouve que cette dernière fais appelle à une autre méthode c'est la méthode « CreateDerectory » qui est utilisé pour créer le repertoire dans l'endroit spécifié.
- Après la création du répertoire il faux le remplir, pour cela on va créer le fichier « main.xml » en utilisant le DOM comme précédemment.

Et maintenant pour terminer toutes les taches qu'on doit les effectuer sur le nouveau projet, on définit la classe « Test » qui utilise la méthode « layout ».

- Dans cette méthode on a fais tous les tests nécessaires sur les données choisi par le client c a d utilisé les classes (MainActivity.java,activitymain.xml, et ajouter l'image) approprier à chaque récupération de données voici un exemple de la classe « Test » :

```
if (sendmessengerie.equals("1") && sendrepertoire.equals("0") &&
sendcalendrier.equals("0") && sendcalculatrice.equals("0") &&
sendgallery.equals("0") && sendmms.equals("0") &&sendphone.equals("0")
&& sendbrowser.equals("0") ) {
    ModifyMainXml.dom ("messengerie");
    ModifyMainJava.set ("messengerie","email");
    AddDrawable.create (Serveur.cheminFichiers+"image1.jpg");
}
```

III-1.4 La compilation du projet :

Après toutes les étapes qu'on a fait, finalement, on a obtenu le nouveau projet qui est prêt à être compiler.

- Pour effectuer cette tache on va utiliser un fichier batch (*.bat) qui contient la commande DOS²³ suivantes : **ant debug** (b) (d)
- ce fichier sera utilisé dans la classe « Compilation.java » (k)qui utilise la méthode « apk() » ,en faisant l'appelle comme suit :

```
cmd = "cmd.exe /c"+Serveur.cheminFichiers+"compile.bat+creerProjet.nom;
```

Mais pour pouvoir utiliser cette commande là, il nous faut tous d'abord télécharger et installer le « **Ant Apache** »

Ant Apache : est un projet du groupe Apache-Jakarta. Son but est de fournir un outil écrit en Java pour permettre la construction d'applications (compilation de programmes et exécution d'autres tâches).

Ant, comparativement à d'autres outils de type «**make**», a la particularité d'être indépendant de toute plate-forme (Unix, Linux, Windows...). Il devient donc un outil fort intéressant dans le cas où on doit développer une application sur plusieurs plates-formes ou encore migrer une application d'une plate-forme vers une autre, son but est de fournir un outil écrit en Java pour permettre la construction d'applications (compilation, exécution de tâches post et pré compilation, ...). Ces processus de construction d'applications sont très importants car ils permettent d'automatiser des opérations répétitives tout au long du cycle de développement de l'application (développement, tests, recettes, mises en production, ...).

Le site officiel de l'outil Ant est : <http://ant.apache.org/>

Pour obtenir plus de détails sur l'utilisation de l'outil Ant, il est possible de consulter la documentation de la version courante à l'url suivante : <http://ant.apache.org/manual/index.html>

Après télécharger et installer l'outil Ant :

- on ajoute le chemin complet du répertoire **bin** de l'outil **Ant** à la variable système PATH (pour pouvoir facilement appeler Ant n'importe où dans l'arborescence du système)
- on s'assure que la variable **JAVA_HOME** pointe sur le répertoire contenant le **JDK**²⁴

²³ disk operating system

²⁴ **Java Development Kit** désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java

- on crée une variable d'environnement **ANT_HOME** qui pointe sur le répertoire **apache-ant-1.9.4** version créé lors de la décompression du fichier
- il peut être nécessaire d'ajouter les fichiers .jar contenus dans le répertoire lib de l'outil **Ant** à la variable d'environnement **CLASSPATH**.

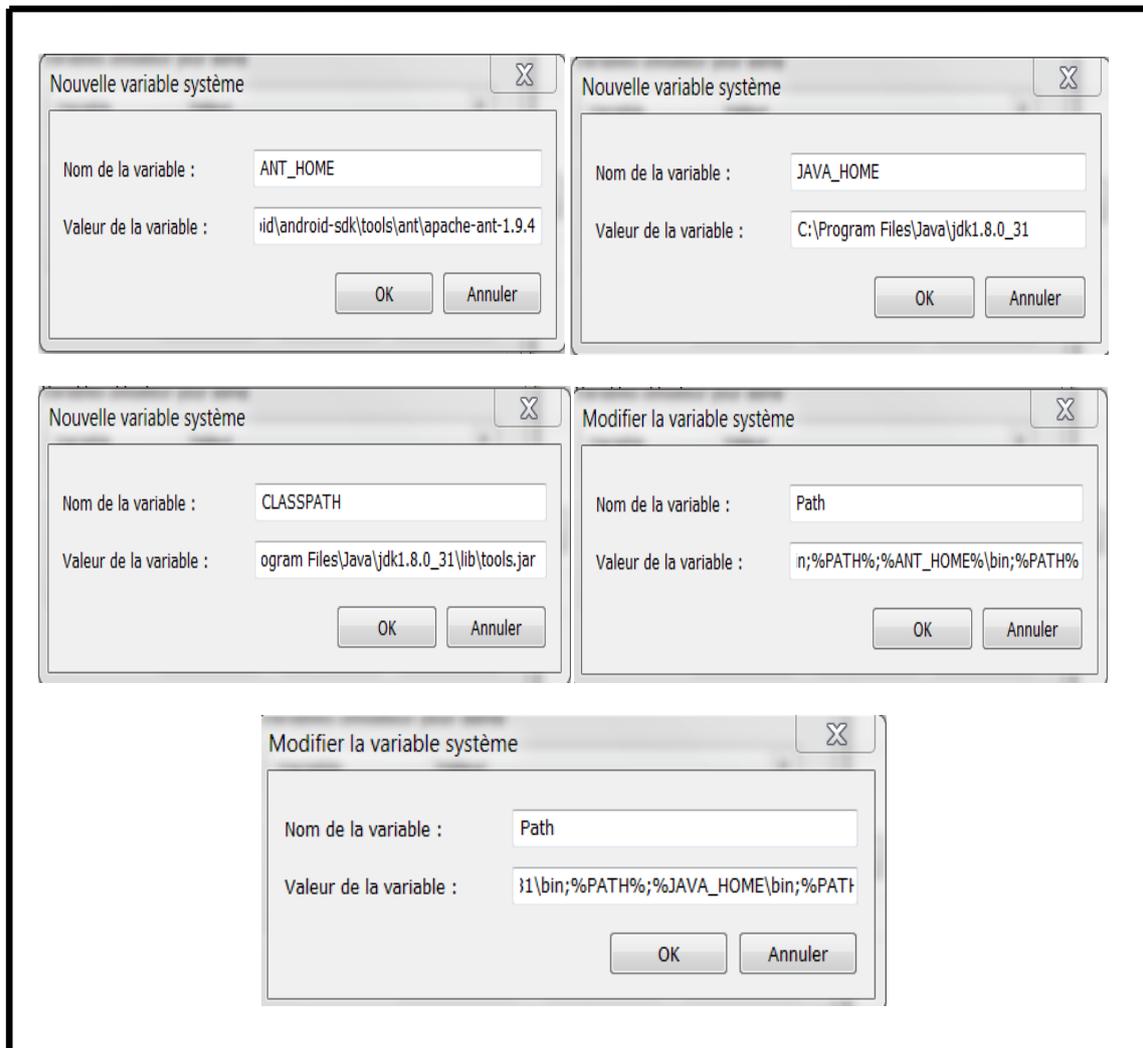


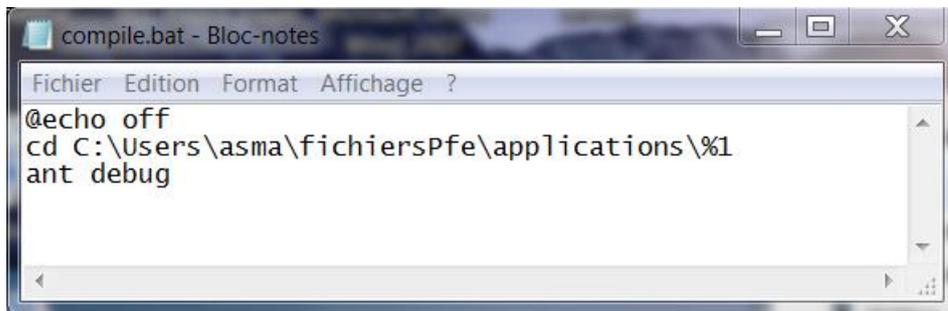
Figure 8: Modification des variables pour utiliser l'Ant apache

Ant s'utilise en ligne de commandes avec la syntaxe suivante :

ant [options] [cible] -> ant debug

Par défaut, **Ant** recherche un fichier nommé **build.xml** dans le répertoire courant. **Ant** va alors exécuter la cible par défaut définie dans le projet de ce fichier build.xml.

Une fois que tout est configuré, on peut accéder au chemin de l'application et la compilé en utilisant un fichier « compile.bat » (e)



```
compile.bat - Bloc-notes
Fichier Edition Format Affichage ?
@echo off
cd C:\Users\asma\fichiersPfe\applications\%1
ant debug
```

Figure 9:Le fichier compile.bat

- La compilation du projet nous permet de générer le APK qu'on va l'envoyer vers le mobile du client, alors si tout va bien, on va trouver le fichier APK et les fichiers de compilation dans le répertoire « bin » du nouveau projet.

III-1.5 Envoyer l'APK

Maintenant passant à l'étape suivante qui consiste à envoyer le fichier vers le client, pour cela on a implémenté la classe « Envoie_apk_generer_socket .java » qui utilise « lancer () » comme méthode. **(i)(j)**

- Dans cette méthode, on va mettre en œuvre une socket, si tout va bien, le client accepte la connexion du serveur, et reçoit un nouveau socket qui est directement lié au mobile, donc il accepte la connexion et il reçoit le fichier APK.
- Et finalement, on ferme les sockets à la fin de l'opération.

Android Client PC Java Server



Figure 10:Connexion client/serveur pour l'envoi du APK

III-1.6 Statistique:

On a implémenté une interface pour notre serveur, cette dernière s'apparaisse lors du l'exécution, en affichant deux boutons, le bouton « **Voir Statistique** » pour voir le tableau de statistique des services les plus, et le bouton « **Voir le graphe** », qui permet d'afficher le graphe correspondant.

Pour pouvoir calculer le pourcentage des services les plus utilisés, on a besoin de recevoir les données de chaque application android , pour cela on crée un fichier xml « **Statistique.xml** » qui contient les applications, ce fichier sera charger dans un tableau.

Le tableau contient les applications par lignes et les services, la description, l'API par colonnes.

On a utilisé deux classes : la classe « **interface.java** »et la classe « **JtableBastique .java** ».

- Dans la classe « **interface.java** », on a déclaré la méthode « **Interface** » qui est utilisé pour créer tous d'abord l'interface graphique en utilisant l'instruction suivante :

```
JPanel pano = new JPanel(new BorderLayout(40,50));
```

- Créer aussi les boutons dans l'interface graphique, le bouton « **Voir statistiques** » et « **voir le graphe** » et puis on affecte une action à chaque bouton.

```
btnConnect.addActionListener(this);  
btnGraphe.addActionListener(this);
```

- Dans la classe « **JtableBasique** », on a déclaré les méthodes « **statistique** », « **stat** », « **JtableBasique** ».
 1. **La méthode statistique** : dans cette méthode on a calculé le pourcentage de chaque service en utilisant le nombre de « 1 » qui apparaisse dans le tableau.
Indiquant que le chiffre « 1 » représente que le service est choisi.
 2. **La méthode stat** : dans cette méthode on s'intéresse au graphe qui représente les statistique des services, alors, tous d'abords,
Il faut crée une vue courbe, pour cela on a utilisé le « **JFreeCharts** » qui nous permet de générer des graphes (courbes) en java.
On a ajouté cette bibliothèque au projet pour pouvoir l'utiliser.
- Notons que JFreeChart est une bibliothèque open source permettent d'afficher des données statistiques sous la forme de graphiques. Elle possède plusieurs formats dont le camembert, les barres ou les lignes ,et propose de nombreuses options de configuration pour personnaliser le rendu des graphiques. Elle permet également d'exporter le graphique sous la forme d'une image. <http://www.jfree.org/jfreechart/>
- On a ajouté les fichiers (.jar) présents dans le répertoire lib décompressé au projet pour pouvoir l'utiliser.
- Une partie "Referenced Libraries" s'ajoute à l'arborescence de notre projet. Ainsi eclipse inclura jfreechart à la compilation du projet.

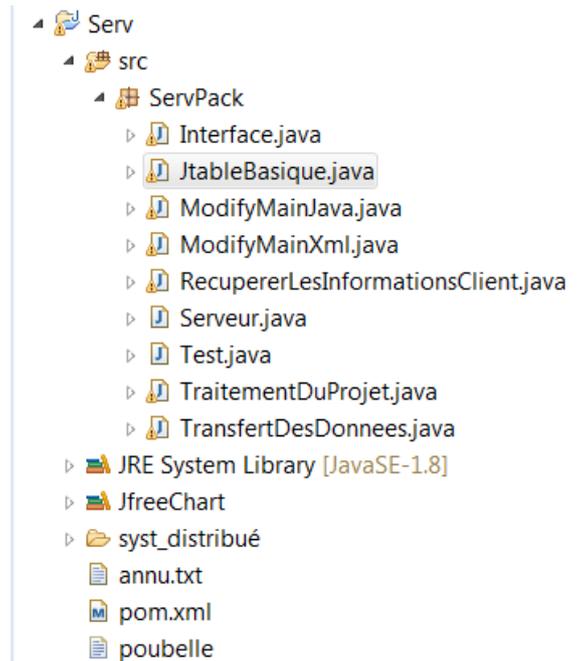


Figure 11:Ajout de la librairie JfreeChart

- Un objet de type JFreechart encapsule le graphique.
- Une instance d'un tel objet est obtenue en utilisant une des méthodes de la classe ChartFactory.

Maintenant la création de la vue s'effectue comme suit :

```
ChartFrame frame=new ChartFrame("First",chart) ;  
frame.pack();  
frame.setVisible(true);
```

Et puis, on a joué sur la totalité du graphe, en ajoutant des couleurs, en changeant le type du texte, et en modifiant aussi la forme du graphe.

On a affiché le pourcentage de chaque service comme suit :

```
JFreeChart chart=ChartFactory.createPieChart3D( "Statistiques sur les  
services", data, true/*legend?*/,true/*tooltips?*/, false/*URLs?*/);
```

```
plot.setLabelGenerator(new StandardPieSectionLabelGenerator("{0}={2}",new  
DecimalFormat("#0.00"),new DecimalFormat("#0.00%")));
```

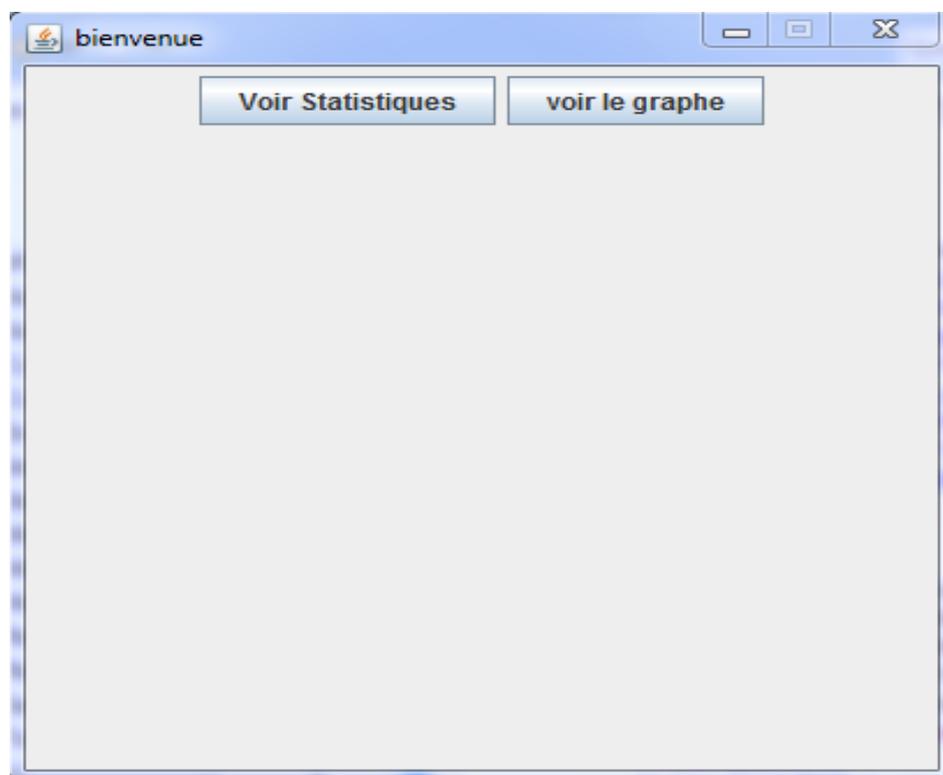


Figure 12:L'interface graphique du serveur

Nom...	API	desc...	calcu...	calen...	mes...	reper...	mms	telep...	galerie	navig...
app	0		0	1	0	0	0	0	0	0
app	0		1	1	1	0	0	0	0	0
app	0	servi...	0	0	0	1	0	1	0	0
app	0	servi...	0	0	0	1	0	1	0	0
app	0	servi...	0	0	0	1	0	1	0	0
app	0	servi...	0	0	0	1	0	1	0	0
app2	0	servi...	0	1	0	1	0	1	0	0
app2	0	servi...	0	1	0	1	0	1	0	0
app3	0	servi...	0	1	0	1	0	1	0	0
app4	0	servi...	0	1	0	1	0	1	0	0
app5	0	servi...	0	1	0	1	0	1	0	0
app6	0	servi...	0	1	0	1	0	1	0	0

Figure 13:Le tableau des statistiques

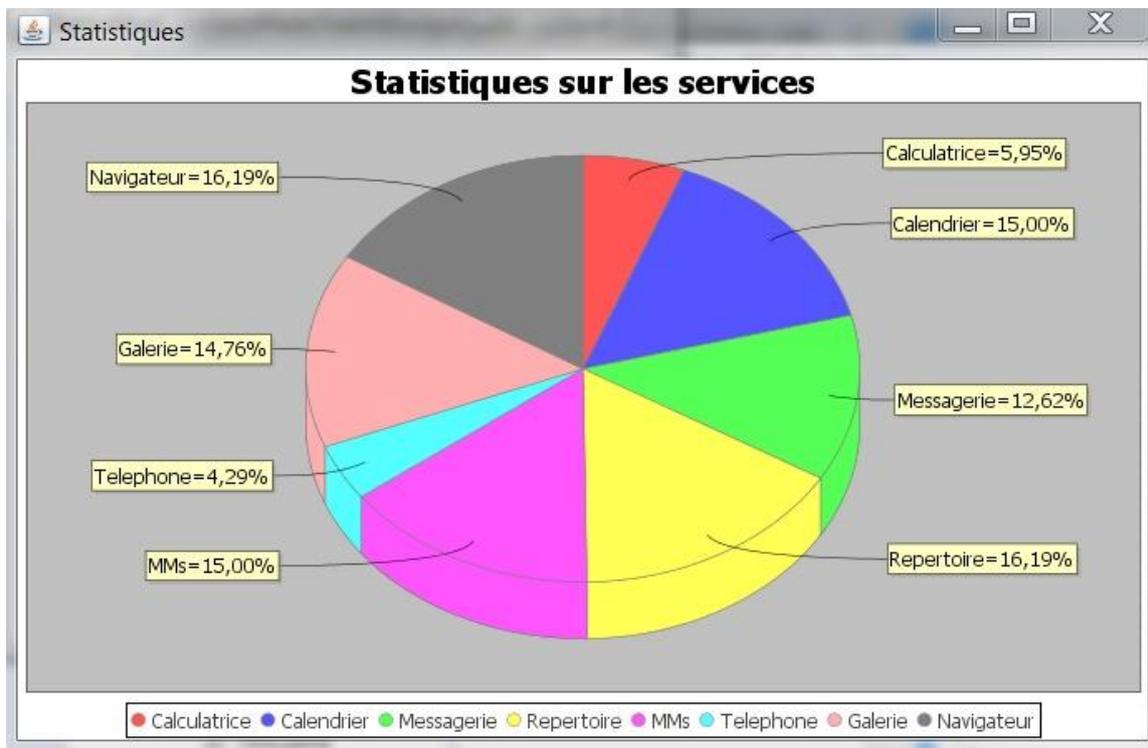


Figure 14:Statistiques des services.

III-1.6 La classe main du serveur :

Maintenant passant à la classe principal du notre serveur c'est la classe « Serveur » qui représente le Main du projet java.

Dans cette classe on fait appelle à toutes les autre classes implémentés précédemment, pour les exécuter, car classe « Serveur » c'est la classe qui contiens la méthode main.

Ici on est arrivé à finir tous les traitements du notre serveur.

Conclusion générale

Ces dernières années, la téléphonie mobile a été sans doute le secteur le plus dynamique, le plus rentable et le plus innovant de toute l'Industrie des Télécommunications.

Le marché des Smartphones, essentiellement, connaît un véritable essor dans lequel les acteurs habituels essaient de s'engouffrer.

Google, ayant réalisé le potentiel de ce marché, a décidé de s'y introduire en sortant un nouveau système d'exploitation Android.

Dans le cadre de notre projet de Voie d'Approfondissement Réseaux et Services Mobiles, nous étions menées à explorer ce nouveau système d'exploitation pour mobiles et à créer une application via une autre application android.

La première étape a été la création de notre propre application android qui représente la partie client.

Dans un second temps, nous avons ajouté des fonctionnalités à cette première application, c'est-à-dire ajouter des boutons pour l'historique, le mail et aussi un site web propre à nous. Et le plus important, c'est d'ouvrir la connexion du client avec le serveur (la deuxième partie) via un bouton « Connecter ».

Dans la deuxième partie, nous avons créé un autre projet java, qui représente le serveur.

La première étape dans cette partie c'est de recevoir les informations envoyées depuis le client, puis les utiliser pour créer la nouvelle application android.

La deuxième étape c'est d'envoyer cet APK vers le client en utilisant les sockets comme outil de connexion.

Dans la dernière partie, on a installé l'APK dans le mobile du client.

Enfin, il faut noter que comme tous les projets, celui-ci est destiné à évoluer au fil du temps.

Perspectives

Notre travail est terminé avec succès, on a pu créer une application android qui donne la main à l'utilisateur de créer une autre application android via son propres Smartphones ou tablette.

Et donc, dans l'avenir proche, nous envisageons d'améliorer notre application, non seulement, on ajoutant de nouvelles fonctionnalités, mais aussi de l'a metre en ligne avec ce qu'on appelle le cloud, tous ça pour facilité la vie d'utilisateur.

ملخص

خلال مشروعنا تم التحدث عن مجال التطبيقات النقالة و كان هدفنا الرئيسي إنشاء تطبيق اندرو يد عبر تطبيق اندرو يد آخر.

وقد تم تطوير العديد من التطبيقات في مختلف مجالات الروبوت .وحاولنا استخدام أحد من هذه التطبيقات لخلق آخر . كانت خصوصية عملنا توليد نموذج التطبيق الجديد وإرسالها إلى الخادم، للحصول على أل (APK)الذي سيبحث إلى الجهاز المحمول.

Résumé

Tout au long de ce modeste travail, nous avons abordé le domaine de développement d'application mobile. Notre objectif principal, était de réaliser une application Android via une autre application Android.

Beaucoup d'applications ont été développées dans divers domaines de l'Android. Cependant nous avons pensé comment utiliser une de ces application pour crée une autre. La spécificité de notre travail était dans la manière de générer le modèle de la nouvelle application et l'envoyer vers le serveur, pour avoir le (.APK) qui va s'installer dans l'appareil mobile .

Abstract

Throughout this modest work, we approached the mobile application development field. Our main goal was to create an Android application via another Android application.

Many applications have been developed in various fields of Android. However we thought how to use one of these application creates another.

The specificity of our work was in the way of generating the model of the new application and send it to the server, to have (.APK) that will move into the mobile device.

Liste de figure

<i>Figure 1 : Les composants d'android (m)</i>	<i>9</i>
<i>Figure 2: Les étapes de conception du projet</i>	<i>13</i>
<i>Figure 3:Le diagramme de cas d'utilisation.....</i>	<i>14</i>
<i>Figure 4:diagramme IDM (l)</i>	<i>15</i>
<i>Figure 5 : Modification de la variable ANDROID HOME.....</i>	<i>18</i>
<i>Figure 6:Modification de la variable PATH.....</i>	<i>18</i>
<i>Figure 7:Arborescence d'un nouveau projet Android.....</i>	<i>19</i>
<i>Figure 8: Modification des variables pour utiliser l'Ant apache</i>	<i>24</i>
<i>Figure 9:Le fichier compile.bat</i>	<i>25</i>
<i>Figure 10:Connexion client/serveur pour l'envoi du APK</i>	<i>26</i>
<i>Figure 11:Ajout de la librairie JfreeChart</i>	<i>28</i>
<i>Figure 12:L'interface graphique du serveur</i>	<i>29</i>
<i>Figure 13:Le tableau des statistiques.....</i>	<i>30</i>
<i>Figure 14:Statistiques des services.</i>	<i>30</i>

Bibliographie

- (a). Récupéré sur http://androidexample.com/XML_Parsing_-_Android_Example/index.php?view=article_discription&aid=69&aaid=94
- (b). Récupéré sur <http://www.androidengineer.com/2010/06/using-ant-to-automate-building-android.html>
- (c). Récupéré sur http://www.java2s.com/Tutorial/Java/0180__File/FileOutputStream.htm
- (d). Récupéré sur - <http://stackoverflow.com/questions/29192720/how-to-compile-android-projects-with-windows-command-line-using-script/29193750#29193750>
- (e). Récupéré sur <http://www.developpez.net/forums/d856423/java/general-java/langage/lancer-bat-partir-d-programme-java/>
- (f). Récupéré sur <http://pastebin.com/6Lu9hkpk>
- (g). Récupéré sur <http://stackoverflow.com/questions/21252879/android-device-connection-with-localhost-server/21253020#21253020>
- (h). Récupéré sur <http://lakjeewa.blogspot.com/2012/03/simple-android-application-to-send-file.html>
- (i). Récupéré sur <https://www.youtube.com/watch?v=foZFmzM1JUk>
- (j). Récupéré sur <http://research.jacquet.xyz/teaching/java/xml-dom/>
- (k). Récupéré sur <http://pastebin.com/6Lu9hkpk>
- (l). Récupéré sur <http://www.lifl.fr/~dumoulin/presentations/cupOfSciences-2010-05-v0.pdf>
- (m). Récupéré sur <http://www.lifl.fr/~dumoulin/enseign/pje/cours/3.modeles/1.introIDM.pdf>